;login:

The USENIX Association Newsletter

CONTENTS

The closing date for submissions for the next issue of *;login:* is August 22, 1991.

THE PROFESSIONAL AND TECHNICAL UNIX® ASSOCIATION

The USENIX Association is a not-for-profit organization of those interested in UNIX[†] and UNIX-like systems. It is dedicated to fostering and communicating the development of research and technological information and ideas pertaining to advanced computing systems, to the monitoring and encouragement of continuing innovation in advanced computing environments, and to the provision of a forum where technical issues are aired and critical thought exercised so that its members can remain current and vital.

The officers of the Association are:

President	M. Kirk McKusick
	mckusick@usenix.org
Vice-President	Michael D. O'Dell
	mo@usenix.org
Secretary	Rob Kolstad
	kolstad@usenix.org
Treasurer	Sharon Murrel
	eowyn@usenix.org
Directors	Rick Adams
	rick@usenix.org
	Ed Gould
	ed@usenix.org
	Evi Nemeth
	evi@usenix.org
	Barry Shein
	bzs@usenix.org
Executive Director	Ellie Young
	ellie@usenix.org

USENIX Association Office
2560 Ninth Street, Suite 215
Berkeley, CA 94710
(415) 528-8649
office@usenix.org

The editorial staff of *;login:* is:

Editor	Ellie Young
Managing Editor	Carolyn S. Carr
Copy Editors	Michelle Dominijanni
	Doug Nielson
Advisor	Rob Kolstad

[†]UNIX is a registered trademark of UNIX System Laboratories.

Judith F. DesHarnais, Conference Coordinator
USENIX Conference Office
22672 Lambert Street, Suite 613
El Toro, CA 92630
(714) 588-8649
judy@usenix.org

Daniel V. Klein, Tutorial Coordinator
(412) 421-2332
dvk@usenix.org

Contributions Solicited

Members of the UNIX community are encouraged to contribute articles to *;login:*. Contributions may be sent electronically to *login@usenix.org* or through the U.S. mail to the Association office. The USENIX Association reserves the right to edit submitted material.

UUNET Subscriptions

UUNET Technologies, Inc.
3110 Fairview Park Drive, Suite 570
Falls Church, VA 22042
(703) 876-5050
uunet-request@uunet.uu.net

Supporting Members

Aerospace Corporation
AT&T Information Systems
Digital Equipment Corporation
Frame Technology, Inc.
Quality Micro Systems
Matsushita Graphic Communication Systems, Inc.
mt Xinu
Rational Corporation
Open Software Foundation
Sun Microsystems, Inc.
Sybase, Inc.
UUNET Technologies, Inc.

USENIX   SERC   IEEE COMPUTER SOCIETY

Call for Participation
Symposium on Experiences with Distributed
and Multiprocessor Systems (SEDMS III)
March 26–27, 1992, Newport Beach, California

In association with:
The Software Engineering Research Center (SERC)

In cooperation with:
ACM SIGARCH, SIGCOMM, SIGOPS and SIGSOFT (Pending)
IEEE-CS Technical Committees on Distributed Processing,
Operating Systems, Software Engineering, and Design Automation

The goal of this symposium is to bring together individuals who have built, are building, or will soon build distributed and multiprocessor systems. SEDMS III will provide a forum for individuals to exchange information on their experiences, both good and bad, including experiences with coding aids, languages, debugging and testing technology, reuse of existing software, and performance analysis. The presentations should emphasize the lessons learned from use of such systems and tools.

Extra-long breaks between sessions and work-in-progress presentations will be provided to facilitate a workshop-like atmosphere during parts of the symposium. We will also have discussion panels on submitted themes.

Six copies of each submission or panel proposal should be sent to the program committee chair (address below) to arrive no later than **November 1, 1991**. Submissions of *full papers* are invited on any topics related to the theme of the symposium. The committee will give preferential consideration to submissions describing experiences with actual systems—*papers describing purely theoretical work will not be accepted*. Panel proposals should include a description of the relevance to the goals of the SEDMS, and the qualifications of the participants suggested.

Important Dates

Submissions due	November 1, 1991
Notifications mailed	December 20, 1991
Camera ready copy due	January 24, 1992

For further information, contact:

General Chair
George Leach
AT&T Paradyne
MS LG-129 Dept. of Computer Sciences
PO Box 2826
Largo, FL 34649–2826
(813) 530–2376
reggie@pdn.paradyne.com

Program Chair
Gene Spafford
Software Engineering Research Center
Dept. of Computer Sciences
Purdue University
W. Lafayette, IN 47907-1398
(317) 494-7825
spaf@cs.purdue.edu

Program Commitee:

John Barr *Motorola*	Sue LoVerso *Encore*
David Cohn *Univ. of Notre Dame*	Dan Duchamp *Columbia Univ.*
Yousef Khalidi *Sun*	John Nicol *GTE Labs*
Andrzej Goscinski *Univ. of New South Wales*	Bharat Bhargava *Purdue Univ.*
Mike O'Dell *Bellcore*	Marc Pucci *Bellcore*
Satish Tripathi *Univ. of Maryland*	Elie Milgrom *Louvain Univ., Belgium*
Howard Katseff *AT&T*	William Bain *Intel*
Thomas Wilkes *Univ. of Lowell*	Karsten Schwan *Georgia Tech*

Call for Papers
USENIX Winter 1992 Technical Conference
San Francisco Hilton, San Francisco, California, January 20–24, 1992

Tutorial Program
Monday and Tuesday, January 20–21

Refereed Papers and Invited Talks
Wednesday through Friday, January 22–24

Some believe that UNIX standardization efforts have killed innovation. And yet, we need innovation, and opportunity for it abounds.

Large write-once disks make the current file-system untenable. Even the 2 gigabyte file limit built in all through the system breaks. Gigabit networking clogs an I/O model designed to push hundreds of kilobytes per second, not hundreds of megabytes. System administration for thousands of machines? Programming tools for distributed workgroups? Object-oriented and visual programming? Microkernels with client/server architectures? RAID disk arrays? Transcontinental file servers? What's a programmer to do?

The USENIX Winter 1992 Conference solicits new work on all topics related to UNIX or UNIX-inspired systems programming and technology. But as always, we care most about innovation and how it coexists with (and sometimes thrives on) stasis.

Please target a sophisticated technical audience, particularly knowledgeable of operating system issues and keenly interested in new and exciting projects in many areas. Vendors are encouraged to submit technical presentations on products. However, we will reject obvious product announcements. Previously published papers will also be rejected, although "retrospective" papers may describe work done years ago.

Submissions must be in the form of extended abstracts,1500–2500 words in length (9000–15000 bytes or 3–5 pages). Shorter abstracts will not give the program committee enough information to judge your work fairly and, in most cases, this means your paper will be rejected. Longer abstracts and full papers simply cannot be read by the committee in the time available. However,

you may append a full paper to an extended abstract; this is sometimes useful during evaluation.

The extended abstract should represent your paper in "short form." The committee will want to see that you have a real project, that you are familiar with other work in your area (i.e., include references), and that you can clearly explain yourself. Please, this is not a mystery to be solved: you should have results and they should be summarized in your abstract. A good submission will contain:

Abstract
* The abstract should be included verbatim in the final paper.

Introduction
* Introduce the problem: why is it important?
* Reference previous work.

How We Solved the Problem
* More details on the problem and its issues.
* Design decisions and tradeoffs, and why they were made.
* Implementation details.

Evaluation
* Data on performance and effort required.
* How well does it work?
* What would you do differently?
* If it failed, why?
* What did you learn from it?

Conclusion
* Summarize the paper, emphasizing why it is important and what was learned.

In addition to the extended abstract, every submission should include:
* A clearly designated contact author who will be your link to the program committee.
* A daytime phone number (essential!).
* A surface mail address (required).
* An email address, if available; email is by far our best path of communication.
* A home phone number (optional, although questions often arise on evenings and weekends and it will avoid delays).

- A FAX number (optional).
- Any special audio/visual equipment you may require. A microphone, overhead projector, and 35mm projector will be provided as standard equipment. We are happy to provide additional assistance and equipment to make your presentation as audio and visually appealing as possible.
- Indication of student status

Presentations are usually scheduled for 25 minutes.

The final date for submissions is August 19. Authors of accepted submissions will be notified by October 1. They will immediately receive instructions for the preparation of camera ready final papers to be published in the conference proceedings. Camera-ready papers of 8–12 typeset pages will be due by November 22.

Submissions can be sent (in order of committee preference):
via email to: SFusenix@usenix.org or uunet!usenix!SFusenix

via paper to:
Eric Allman
Computer Science Division, EECS
University of California
Berkeley, CA 94720

via FAX to: (415) 843–9461

Awards for Best Papers

A cash prize for the best paper by a full-time student will be awarded by the conference program committee. With your submission, please indicate if you are a full-time student.

An award for Best Overall Paper at the conference is also made by the committee.

TECHNICAL PROGRAM COMMITTEE

Chair: Eric Allman, *University of California, Berkeley*
Rick Adams, *UUNET Technologies, Inc.*
Andrew Birrell, *Digital Equipment Corporation, Systems Research Center*
Tom Ferrin, *University of California, San Francisco*
Bob Gray, *US West Advanced Technologies*
Teus Hagen, *OCE*
Steve Johnson, *Athenix*
Pat Parseghian, *AT&T Bell Laboratories*
Dennis Ritchie, *AT&T Bell Laboratories*
Greg Rose, *IBM Thomas J. Watson Research Center*
David Rosenthal, *Sun Microsystems*
Brent Welch, *Xerox PARC*

RELEVANT DATES

Abstracts Due
Monday, August 19
Notification to Authors
Tuesday, October 1
Camera-ready Papers Due
Friday, November 22

Materials containing all details of the technical and tutorial program, conference registration, hotel and airline reservation information will be mailed in October 1991. Contact the USENIX Conference office.

Computing Systems Correction

Volume 3, number 4 (Fall 1990) of Computing Systems was inadvertently printed with the wrong table of contents on the back cover. The journal's publisher, the University of California Press, has produced a sticky label with the correct contents that can be placed on the back cover.

If you would like to receive one of these labels, please send your name and postal service address to the following email address: *journal@garnet.berkeley.edu*

Please allow two weeks for delivery.

Preliminary Announcement
USENIX Mach Symposium, Monterey, California
November 20–22, 1991

Background

Mach has become a dynamic addition to the operating systems marketplace. DARPA originally sponsored Mach development, and continues to emphasize the use and growth of Mach. In the larger research community, Mach is ever more widely used at many university sites and industrial research labs. Versions of Mach have been released commercially by Encore, NeXT, BBN, and mt Xinu. The Open Software Foundation chose Mach as the basis for its operating system offering; now, OSF/1 is finding increasing acceptance as computer vendors ready products derived from it.

Recent developments have demonstrated the feasibility of Mach 3.0, the combination of a pure Mach kernel with single or multiple servers emulating the features of traditional operating systems. Performance of Mach 3.0 has begun to approach or exceed that of Mach 2.5. Workers outside of the CMU community have begun to use Mach 3.0 as the basis for their projects. In short, acceptance of Mach has come about in an astonishingly brief time.

Activity in this field has been sufficiently widespread that, little more than a year after the first USENIX Mach workshop, the USENIX Association is pleased to sponsor an expanded Mach symposium to bring together researchers, engineers, vendors, and users of Mach systems. We will encourage discussion of all past and present Mach-related research, development, production, and applications activities.

Symposium Overview

The symposium will be spread over three days. The first day will be devoted to two half-day tutorials on advanced programming for Mach 3.0. The following two days will concentrate on presentation of refereed papers on current and historical Mach-related work. Long breaks between presentations provide ample opportunity for informal discussion. Some time will be available for descriptions of work in progress.

Tutorials

Richard Draves *Writing a Multi-Threaded Mach 3.0 Server*
David Black *Writing an External Memory Manager*

Richard Draves will lead a tutorial analyzing the process of writing a multi-threaded server, with particular attention paid to the complexities of using Mach IPC. During the course of his doctoral studies at Carnegie Mellon University, Rich rewrote Mach 3.0 IPC to solve problems that became apparent with Mach 2.5 servers.

David Black will demonstrate how to create an external memory manager; discussion will center on the intricacies of developing an efficient (and well-behaved!) external manager. David, currently of the Open Software Foundation, received his doctorate from Carnegie Mellon for his contributions to Mach.

These tutorials are being developed especially for this USENIX Mach symposium. They will explore concepts and rationale as well as real examples. They are oriented towards programmers who already have some familiarity with using Mach IPC and VM. Each tutorial is a half-day, so conference attendees may take part in both. The tutorials will be priced separately from the conference registration fee.

For further information about registration for tutorials, technical sessions, and hotels contact the USENIX Conference office.

Program Committee

Alan Langerman,
Encore Computer Corporation....Program Chair
Larry Allen, *Open Software Foundation*
Nawaf Bitar, *Hewlett-Packard Company*
Susan LoVerso, *Encore Computer Corporation*
Melinda Shore, *Cornell University*
Michael Young, Ph.D., *Transarc*

Announcement and Call for Papers
CONVENTION UNIX 92
5th Conference and Exhibition
CNIT, Paris - La Defense March 23–27, 1992

Open Systems: Inter-operability

The Associaton Francaise des Utilisateurs d'UNIX et des Systemes Ouverts (AFUU) is organizing, in collaboration with the Bureau International de Relations Publiques (BIRP), CONVENTION UNIX 92 from 23rd to 27th March, 1992, at CNIT, Paris-La Defense.

This event is centered on tutorials and conferences dedicated to UNIX and Open Systems. A display of hardware and software, organized by more than 150 exhibitors, will take place at the same time.

CONVENTION UNIX 92 is an even more important event than usual. 1992 will provide the occasion for us to celebrate the tenth anniversary of AFUU.

The Theme

Users have more and more sophisticated operational requirements and the facts prove that these needs can no longer be satisfied in the sole context of proprietary products. Users want open and evolutionary systems which make the best use of the variety of solutions offered to them.

Software and interoperability is one solution. It may be defined as the ability with which applications and systems of different origins to cooperate to produce an integrated service for the user. In the same way that we have been able to come to terms with the hardware heterogeneity, interoperability enables bridges to be built between different types of software components in order to develop fully open systems.

The purpose of CONVENTION UNIX 92 is to clarify and illustrate this notion of interoperability, both from the technical and economic points of view. What does it provide to the users and to the companies? What are the benefits for the users? What are the economic benefits? How is it done? What are the concrete and demonstrable examples of it?

The Programme Committee would like to receive submissions in the following three categories:

1) Man/machine interfaces:

They provide the user with an ultimate view of the realization of his needs. Hence, man/machine interfaces are one of the keys to interoperability. By facilitating communication with the machine (integration of voice, data and picture), they should lead to better productivity and to "interoperability" between men and machines. Although the ergonomics and operator comfort of existing tools are now an established fact, the variety of interfaces still remains a barrier to be overcome.

2) Applications:

Applications are the essential components in the creation of services (data management, transactional processing/OLTP). How do they cooperate? Are the emergence and the adherence to standards, and API definitions, sufficient to guarantee the interaction between components? How can compatibility be defined? Does compatibility between interfaces guarantee the portability of applications? How can we reuse these components to improve return on software investment?

3) The tools of inter-operability:

What are the bases on which interoperability between men, interfaces, and applications may be founded? What are the basic services necessary to the applications and to the man/machine interfaces to permit cooperation (distributed systems, communications protocols, RPC)? What are the languages and methods which facilitate the implementation of interoperability (programming and object-oriented languages, interface generators)? What are the consequences on software engineering? What are the development costs?

The purpose of this Conference is to create a vast forum in which information can be exchanged about experiences with interoperability,

the technology, the risks and benefits and the applications, in everyday use. The papers may cover users experiences, industrialists' and manufacturers' strategies, technological innovation in the research world, or even the broad principles in a particular field.

Tutorials

The tutorials will provide the listeners with lectures on clearly defined subjects. Their purpose will be to present the state-of-the-art of the most important points in the implementation of interoperability. As was the case last year, these tutorials will be led by experts of national and international fame. Persons interested in lecturing at a tutorial are invited to contact the chairman of the Programme Committee as soon as possible.

Important Dates

September 27, 1991: deadline for the receipt of full papers or extended abstracts by the Convention Secretariat.

October 25, 1991: notification to authors of the Programme Committee's decision.

December 13, 1991: deadline for the receipt of the final texts by the Convention Secretariat.

Submission of Papers

Paper proposals should have a title, the name of the author and the organization to which the author belongs. These proposals should include the complete text (5 to 10 pages) or at least an extended abstract (2 pages). They will be examined on the basis of their quality, originality, and the adherence to the general theme of the Convention (the interoperability of Open Systems) and to the directives given:

Man/machine interfaces: graphics, multimedia, ergonomics, learning;

Applications: DBMS, transactional processing, standards implementation;

Tools: distributed processing, RPC, multipersonality systems, communication between UNIX and other systems, object-oriented languages, software engineering.

The Programme Committee wishes to include both technical papers and syntheses of different approaches. The quality of the lectures depends to a large extent on the facilities which are provided to the Programme Committee to enable it to make its decisions on the acceptance of a paper. Consequently, candidates are strongly encouraged to supply a complete version of their text as soon as possible.

Official languages are French and English. A simultaneous translation service will be available for conferences and tutorials.

Proposals should be sent to:
A.F.U.U.
Secretariat de la CONVENTION UNIX 92
11, rue Carnot
94270 Le Kremlin-Bicetre - Fran ce
Telephone : (+33) 1.46.70.95.90
Fax : (+33) 1.46.58.94.20
Telex : 263887 F
E-mail : *afuuconf@inria.inria.fr*

Convention Chairman
Sylvain Langlois
Electricite de France - Direction des Etudes et Recherches
Tel : (+33) 1.47.65.44.02
Fax : (+33) 1.47.65.30.01
E-mail : *sylvain@cli53an.edf.fr*

Programme Committee Chairman
Francois Armand
Chorus systemes
Tel : (+33) 1.30.64.82.36
Fax : (+33) 1.30.57.00.66
E-mail : *francois@chorus.fr*

Calendar of UNIX Events†

1991 Sept 10–12	Sun User Group	Birmingham, UK
1991 Sept 16–20	EurOpen	Budapest, Hungary
1991 Sept 24–27	AUUG	Sydney, Australia
1991 Sept 30–Oct 3	*LISA V	San Diego, CA
1991 Oct 7–11	Interop	San Jose, CA
1991 Oct 10–11	Multi-User	UniForum Montreal, Canada
1991 Oct 21–25	IEEE 1003	Parsippany, NJ
1991 Oct 3–Nov 1	UNIX EXPO	NY, NY
1991 Oct 31	Sun User Group	The Netherlands
1991 Nov 4–8	ISO/IEC CJT1 SC22 WG15	Stockholm, Sweden
1991 Nov 14	APP/OSE Users Forum	NIST, Gaithersburg, MD
1991 Nov 14–15	JUS Symposium	Osaka, Japan
1991 Nov 20–22	*Mach Symposium	Monterey, CA
1991 Nov 27–29	AFUU	Grenoble, France
1991 Dec 3–4	UNIX Fair '91	Tokyo, Japan
1991 Dec 7–10	Sun User Group	San Jose, CA
1991 Dec 9–13	DECUS	Anaheim, CA
1991 Dec 16–18	UKUUG	Edinburgh, UK
1991 Jan 13–17	IEEE 1003	Irvine, CA (location tentative)
1992 Jan 20–24	USENIX	San Francisco, CA
1992 Jan 22–24	UniForum	San Francisco, CA
1992 Mar 4–7	Computers in Libraries	Westport, CT
1992 Mar 11–18	CeBIT 92	Hannover, Germany
1992 Mar 24–27	AFUU	Paris, France
1992 Mar 26–27	*SEDMS III	Newport Beach, CA
1992 Apr 6–9	EurOpen	Jersey, UK
1992 Apr 6–10	IEEE 1003	Atlanta, GA
1992 May 4–8	DECUS	Atlanta, GA,
1992 May 18–22	ISO/IEC CJT1 SC22 WG15	
1992 Jun 2–4	UNIX EXPO West	Anaheim, CA
1992 Jun 8–12	USENIX	San Antonio, TX
1992 Jun 22–24	Sun User Group	Washington, DC
1992 Summer	UKUUG	Belfast, Northern Ireland
1992 Jul 13–17	IEEE 1003	
1992 Autumn	ISO/IEC JTC1 SC22 WG15	Denmark
1992 Sept 8–11	AUUG	Melbourne, Australia
1992 Oct 6	ISO/IEC JTC1 SC22 WG15	Denmark
1992 Oct 19–23	IEEE 1003	
1992 Oct 26–30	Interop	San Francisco, CA
1992 Nov 25–29	EurOpen/UniForum	Amsterdam, The Netherlands
1992 Dec	UKUUG	Manchester, UK
1993 Jan 11–15	IEEE 1003	
1993 Jan 25–29	USENIX	San Diego, CA
1993 Mar 16–18	UniForum	San Francisco, CA
1993 Mar 24–31	CeBIT 93	Hannover, Germany
1993 Apr 5–19	IEEE 1003	
1993 Apr 26–30	EurOpen	
1993 Jun 21–25	USENIX	Cincinnati, OH
1993 Oct 25–29	Interop	San Francisco, CA
1993 Autumn	EurOpen/UniForum	Utrecht, The Netherlands
1994 Jan 17–21	USENIX	San Francisco, CA
1994 Mar 16–23	CeBIT 94	Hannover, Germany
1994 Mar 23–25	UniForum	San Francisco, CA
1994 Jun 6–10	USENIX	Boston, MA
1994 Sep 12–16	Interop	San Francisco, CA
1995 Jan 16–20	USENIX	New Orleans, LA
1995 Feb 21–23	UniForum	Dallas, TX
1995 Jun 19–22	USENIX	San Francisco, CA

†Compiled with the assistance of Susanne Wilhelm of Windsound Consulting.
*USENIX workshops, symposia, and mini-conferences

EurOpen Publications

The following publications are available from EurOpen. If enough USENIX members express an interest, we may make these publications available through our office. Send email to *office@usenix.org* if you would be interested in such an arrangement.

Proceedings:

Dublin	Autumn '83	$ 3.28
Nijmegen	Spring '84	8.20
Cambridge	Autumn '84	8.20

Paris	Spring '85	8.20
Copenhagen	Autumn '85	16.40
Finland/Sweden	Spring '87	2.80
Dublin	Autumn '87	32.80
Munich	Spring '90	32.80
Nice	Autumn '90	41.00
Tromso	Spring '90	41.00

EurOpen Email Directory, 2nd edition   32.80

(List prices are given in U.S. dollars, based on current exchange rates.)

1992 USENIX Nominating Committee

The Nominating Committee for the USENIX Association board of directors is comprised of:

Marc D. Donner, *IBM Research*
Andrew Hume, *AT&T Bell Labs*

Peter H. Salus, *Sun User Group,* chairman
Charles Sauer, *Dell Computer*
Elizabeth Zwicky, *SRI International*

1991 SUG CD-ROM

The Sun User Group and Young Minds are proud to announce the availability of the 1991 SUG CD-ROM. Over 500 MB of source code, software archives, binaries and Sun Microsystems' patches are included in this year's collection.

For more information, please email *office@sug.org* or call 617/232-0514. To order your copy of the CD-ROM, complete the form and return it to the Sun User Group office.

TABLE OF CONTENTS

Top level (mounted under /usr/sug)

README - on line installation directions and overview

TOC	- Table of contents - this file.
archive	- Archives of mailing lists, Sun patches, and other miscellaneous stuff.
bin	- Sun 4 binaries compiled under 4.1.1
doc	- Documentation
etc	- Sun 4 binaries compiled under 4.1.1. Typical kinds of things you would expect to find in /etc.
include	- Include files required to compile or run software found on SUG CD.
lib	- Libraries, etc.
man	- Man pages
share	- Architecture independent files like icons, audio files, images, etc.
src	- source galore

The 1991 SUG CD ORDER FORM

I hereby agree to these conditions:

1. The contents of the archive continue to be owned exclusively by their authors or owners. I simply get a copy for my own use, subject to the restrictions of paragraph three, below.
2. Sun User Group, Inc. (SUG) does not own, and makes no warranty whatsoever as to the contents of the archive, but acts as a collection and copying service only. SUG is not responsible for any damages whatsoever, arising from any use of the contents of the archive.
3. My right to use the contents of the archive is restricted by the expressed wishes of the owners. I will not violate the conditions imposed on such use, such as restriction on further copying or incorporation into a product. These conditions are documented in the _readme files in the archive.

The price of the CD is $250.

Shipping and handling: Add $10 (USA) or $25 (Intl.) If you are not a member of the Sun User Group, add $40 (USA) or $55 (International) to the above sums for 1991 membership. You must be a SUG member to purchase the CD-ROM. I enclose a US $ check for:

_____ $260 (SUG member in the USA)
_____ $275 (SUG member outside the USA)
_____ $300 (Includes membership inside the USA)
_____ $330 (Includes International membership)

Name _____ Signature _____

Company Name _____

SUG Membership #(if known) _____

Electronic Mail Address _____

Telephone Number _____

Check Enclosed ____ Mastercard ____ Visa ____ Credit Card # _____ Exp. date. _____

Card Holder : _____ Signature: _____

Purchase Order # _____

Ship to: Bill to:

_____

_____

_____

_____

OUTSIDE THE U.S. ONLY:

We will be pleased to send you a copy of the archive, but before our government will permit shipment, we must have an originally signed letter of assurance by an authorized employee, on your letterhead, in triplicate.

Individuals outside of the USA may find using their credit cards easier than purchasing US$ checks as this eliminates bank charges.

Archive Committee
Sun User Group   +1 617 232–0514
1330 Beacon Street, Suite 315   Fax: +1 617 232–1347
Brookline, MA 02146   Email: *office@sug.org*

Report from the Summer '91 Conference

Best Paper Award

The Program Committee named Matthew Blaze and Rafael Alonso of Princeton University as the authors of the best student and overall paper. Their paper "Long-Term Caching Strategies for Very Large Distributed File Systems" was presented at the Summer USENIX Conference in Nashville, Tennessee.

Work in Progress Session
Lisa Bloch, Chair

Shigetoshi Yokoyama
Spatial Data Management with Dynamic Icons

Graphical browsing of data has been profound as an interface for understanding the content of large multimedia databases. To do this effectively, there is a need to have a large bandwidth of information under active consideration by the user. We are presently limited in this respect by the size of the workstation display and in the way in which the information is represented.

We present here a tool which allows the creation of a three dimensional workspace which uses the illusion of depth to structure and display information. The tool uses a fast pyramid image coding technique based on wavelet functions to allow rapid smooth zooming of display windows which remain "active" even when collapsed to icon dimensions. We call these intermediate status windows "Dynamic Icons" or DICONs. These DICONs can be applied to general multimedia documents such as images and video.

Authors: Shigetoshi Yokoyama,
Naoya Kawamura (NTTDATA),
John R. Williams, Jerome J. Connor (Civil Engineering Dept. of MIT).
Intelligent Engineering Systems
Laboratory
MIT
77 Mass. Ave., Room 1–250
Cambridge, MA 02139
yoko@iesl.mit.edu

Dan Muntz and Peter Honeyman
CITI, The University of Michigan, Ann Arbor
Trace-driven Analysis of Multi-level Caching in Distributed File Systems

At the IFS Project, we have been investigating the potential for intermediate file servers to address scaling problems in increasingly large distributed file systems. To this end, we have run trace-driven simulations based on data from DEC-SRC and our own data collection to determine the potential for caching-only intermediate servers.

The degree of sharing among clients is central to the potential effectiveness of an intermediate server. This turns out to be quite low in the traces available to us. All told, fewer than 10% of file accesses are to shared files.

Trace-driven simulation shows that even with an infinite cache at the intermediate, cache hit rates are disappointingly low. For client caches as small as 20 MB, we observe hit rates less than 19%. As client caches increase in size, the hit rate at the intermediate approaches the degree of sharing.

Marcus Leech
Bell-Northern Research, Information Technology Division
Managing a UNIX Help Desk-The Help Desk Environment Project

Large corporations (such as BNR) are increasingly reliant upon the services provided by various kinds of technical Help Desks to provide technical and software support to their development communities.

The Software Support group provides software support and consulting to a user population with BNR and Northern Telecom of approximately 10,000 users. The last two years has seen a dramatic shift in the computing environment away from mainframes to various types of UNIX workstation.

The Software Support group has developed, over time, a set of tools useful for the job of providing VM/CMS technical support. With the introduction of UNIX workstations into the environment came an opportunity to develop an "environment" suitable for many of the Help Desks within BNR that use workstations to do their job.

The UNIX consulting team with the Software Support group has embarked on a project, called the Help Desk Environment, to provide a general purpose set of tools for technical Help Desks within BNR. Since the focal point of a Help Desk/ Hotline consulting group is the telephone, the environment is closely integrated with the telephone switching technology that provides automatic distribution of calls to available consultants.

Marcus Leech
4Y11
Bell-Northern Research
P.O. Box 3511, Stn. C
Ottawa, ON, CAN K1Y 4H7
mleech@bnr.ca
VE3MDL@VE3JF.ON.CAN.NA

Danny Chen
Towards a New Interface for Specifying and Accessing Kernel Performance Metrics for Unix System V

The UNIX international performance management working group has been working on the issues that need to be addressed in order to improve the access to useful system and application performance data and increase the availability of performance measurement and analysis tools on UNIX systems.

I will present my perception of some of the ideas that are being discussed at the UI performance management working group meetings as well as discuss details of my prototype implementation of a system that provides a "performance metric registration" facility and user level read-only memory mapped access to kernel performance measures.

This should be of interest to developers of performance tools as well as developers of large applications such as DBMSes who currently find that accessing performance or resource usage information is too expensive to include in production systems.

Geoff Collyer
nam: A Manual Page Decompiler (extracting structure from formatted manual pages)

Now that AT&T is charging a small fortune for manual page source, fewer vendors of UNIX systems ship manual page source, or do so only at exorbitant cost. As a result, even sites that pay for program source code rarely have manual page source. Consequently, even sites with source code now often have out-of-sync manual pages, and this form of one communications medium (structured text) is in decline.

We describe one somewhat bizarre solution to this unacceptable state of affiars, which we hope will help resuscitate manual pages: decompiling or reverse-engineering manual pages: that is, extracting structure from formatted manual pages.

Geoff Collyer
Software Tool & Die
1330 Beacon St. #215
Brookline, MA 02146
geoff@world.std.com
geoff@uunet.uu.net

Brad Templeton
A New Data Compressor/Archiver for UNIX, DOS, and Other Systems

The author will discuss details of a new data compressing and archiving program which provides some of the best available general data compression and is particularly good at compressing USENET news. The archiver is portable and designed to run on different systems, but it supports all Unix features such as links and symlinks.

Brad Templeton is President of ClariNet Communications, which provides an electronic newspaper in USENET format, moderator of the USENET rec.humor.funny newsgroup and author of lots of software.

Rick Macklem, University of Guelph
Not Quite NFS, Crash Tolerant Cache Consistency for NFS

Not Quite NFS (NQNFS) is an NFS-like protocol designed to maintain full cache consistency between clients in a crash tolerant manner. It is an adaptation of the NFS protocol such that the server supports both NFS and NQNFS clients while maintaining full consistency between the server and NQNFS clients. It borrows heavily from work done on Spritely NFS [Srinivasan89], but uses Leases [Gray89] to avoid the need to recover server state information after a crash.
rick@snowhite.cis.uoguelph.ca

Animal in the Safe?

A playful spirit pervaded at the O'Reilly books booth on Thursday afternoon, June 13, the last day of the USENIX exhibition. Peter Collinson had posed in /etc/motd the question: "What is the name of the animal in the safe on the cover graphic of O'Reilly's new *Practical UNIX Security* book?"

Always ready for some fun, the O'Reilly folks turned the question into a contest. After all, no one on the staff knew. Perhaps one of the book's readers would! At 4 o'clock, as the doors closed, Ellie Young and Peter Salus judged the entries and picked the winner: Charles Craig from Northwestern University, who said it was "Schroedinger's cat." The runner-up, Henry Spencer from the University of Toronto, thought it was the "Internet worm (of course)". Two honorable mentions went to Jeff Okamoto of Hewlett-Packard who proclaimed, "a dead one," and Brad Templeton of Looking Glass Software, who named it "George."

Congratulations to Charles, who won the book and a t-shirt and to the runner-up, who got a Practical UNIX Security t-shirt.

The Egregious Use of UNIX Utilities Contest

David Tilbrook, Sietec Open Systems Division

1. It's another fine mess you've got me into Debbie

It began innocently enough.
New mail for dt@nixcan has arrived:
Date: Thu, 30 May 91 01:31:34 EDT
To: dt@snitor.uucp jaap@mtxinu.COM
ed@mtxinu.COM henry@zoo.toronto.edu
shore@THEORY.TC.CORNELL.EDU
Subject: Nashville - important job
From: scherrer@mtxinu.COM
I am looking for a few good people to populate an august panel of judges to make those winning decisions in the "most
...more...
Eagerly I read my mail ...
..."most egregious misuse of a Unix utility contest" ...
... came to mind as ones who would be eminently appropriate ...
... announce the contest ...

... set up a box at the registration desk ...
... have a few drinks at the bar...

My interest was piqued. I had been planning to not drink at this conference, but it seemed that duty called.

"So, would you any and all be willing to serve, and handle this most important of all tasks?"

How could I refuse such a responsibility?
There followed a number of letters discussing rules and prizes.

"I suppose we should wait until getting to Nashville to choose prizes. There ought to be sufficiently tacky things available there." — ed
"And the prize obviously needs to be egregiously inappropriate ..." — melinda
"A free membership of Decus?" — jaap
"Egregious, but DECUS memberships are free, anyway. Maybe a Dolly Parton doll? After all, we want to be sure it's not just memorable, but reminiscent." — ed

It was obvious that the "august panel" had been populated, so three further tasks had to be initiated: determining the prize budget, formulating the contest rules, and generating a few seed entries to encourage the appropriate responses from the contestants. First of all, the budget! At the last USENIX conference at which I was responsible for the prizes, (the Atlanta Errno contest), I had asked Judy for a modest amount of money for prizes for the errno contest and ended up with $200 (which was a order of magnitude more than we had spent in Florence for the same contest). By the way, given the winner was Mark Bartelt, who lives close by in Toronto, I obligated to tell you that $70 bottle of bubbles was delicious. But Mark was not going to be attending Nashville, so an even more modest budget was acceptable.

The rules, close to the form presented at the conference (given in the following section) were composed and accepted — well, no one objected.

Generating seed examples turned out to be an expensive proposition.

Task: generate random number in range 0 to 59
Implementation:
```
date | sed 's/.*:.*:\(..\).*/\1/'
```

was the first. It obviously inspired a handful of random number entries, the worst (i.e., best entry) of which was to sum the file names in /usr/spool/news (from Kenneth Ingham).

The second that came to mind was:

> Task: Provide a useful GUI.
> Implementation: Install X11 ;-)

This seed was used to discourage similar entries, but to no avail:

> Task: Edit a file
> Implementation: emacs
> — Matt Blaze

Naturally, Jaap suggested his *basic* in *troff* implementation, and just as was to be expected there were a number of entries that proposed implementing *troff* using Henry's *awk* implementation and then using Jaap's *basic-in-troff* to implement *basic*. These obviously met the egregious criteria, but lacked originality (see rules).

It was at this time, that Russel Crook (planet of origin unknown) initiated what was to eat up at least 6 person-days and millions of cycles — coming up with really bad ways to reverse a file, such as:

```
x='wc-l<$1'
text $x -eq 0 && exit
y=1
while [ $y -lt $x ] ; do
        tail -$y $1 | sed 1q
        y='expr $y + 1'
done
sed 1q $1
```

and not to be outdone:

```
x=/tmp/split$$
mkdir $x
cat $1 | ( cd $x ; split -1 ; cat 'ls -r' /dev/null)
rm -rf $x
```

Worked a treat, although somewhat slowly, and had trouble with more that 676 (i.e., beyond *xzz*) lines.

```
lc='wc -l<$1'
test $lc -le && cat $1 && exit
tail -'expr \( $lc + 1 \) / 2' $1>.$$
sh $0 ,$$; rm -f ,$$
sed 'expr $lc - \(\( $lc + 1 \)/ 2 \)'q $1>,$$
sh $0 ,$$; rm -f,$$
```

```
#       Might fail due to wrapped pids
#       or running out of processes
```

The use of *wc* in all of the above led to the generation of the following alternative for *wc*:

```
(sed 's/.*//'; echo 'expr _LINE_ - 1') | /lib/cpp
| sh
```

Our experimentation was a success, albeit somewhat costly, but resulted in two pleas for help ...

> "Oh dear, ... how does one turn this off? And how are we to judge this?" — dt

Jaap provided the answer that proved to be prophetic:

> "Just like we did at the errno judging: sit around at a party, having people supply us beer in exchange for the entries we turn down so that they can fall about laughing while we make the short list looking very serious."

He also suggested a minor optimization to one of the file reversing scripts — sometimes he worries me.

2. The Rules

Entries will consist of:

The Task
  A short specification of the task to be performed.
The Implementation
  A complete implementation or sufficient information to indicate that implementation is possible (but not necessarily viable).
Bugs
  If necessary to justify why solution to be considered egregious.

Entries to be submitted on a single A6 sheet (approximately 1/4 letter size for the 'merkins), with submitter's name and company on the back and entries will be accepted up until 17:30, Thursday, June 13th, at the USENIX desk.

2.1. Judging

The winning entry will **probably** be one that is:

- outstanding for the wrong reasons (i.e., egregious),
- amusing,

- original,
- well presented and presentable,
- correctly implemented, but not in a way that anyone unbent would do it, **and**
- was submitted by someone with a demonstrated appreciation of how thirsty the judges get.

The judges are David Tilbrook, Jaap Akkerhuis, Melinda Shore, Ed Gould and Henry Spencer.

2.2 Prizes?

Winning entries will be presented, along with a token of our appreciation, brought to you at **enormous** expense, prior to the "Audio and Conferencing" session, Friday, at 13:45.

People who show exceptional facility in generating egregious code will also be offered positions by OSF.

3. Honourable Mention #1

This entry was awarded an Honourable Mention, because the judges were not brave enough to read it.

Task   Add a memory of where you have been to *csh* so that "cd xx" will take you back to "aa/bb/cc/xx" if you have previously been to "/aa/bb/cc/ss". Add the last component of your path to your prompt of "/" if you are root. Keep a directory stack of where you've been (pushd) but "cd" should take you to your home directory. Print out your new directory iff it could be considered unexpected. Allow popd'ed directories to be unpopped. Do not fork.

Implementation

```
alias pu 'pushd \!*>/dev/null; eval __a\! __d;
eval __f $cwd:t \'
alias cd 'set __d=(\!*);__a $__d[1] ; pushd
$__d[1]>/dev/null; eval __f $cwd:t/'
```

. . . and so on for another 15 lines of aliases, part of which was:

```
'\!:1'\" =~ [A-Za-z_]* && '\"\!:1;\" !~ *~ . . .
'\!:1'\" ! *.a+%\ ,:\$* ) eval . . .
```

Author   David Muir Sharnoff, Comdisco Systems

Readers are invited to communicate with David to receive the entry in its full glory.

4. Honourable Mention — Best use of Egregious Tools

Task   Copy a file between two hosts.

Implementation
    uuencode source. Split into 8 byte chunks. Use adb to write chunks into /etc/utmp, pausing after each write. Use rwho on remote system to retrieve chunks.

Author   David Muir Sharnoff

An honourable mention was awarded, rather than a prize, because the judges were not brave enough to shake hands with the author.

5. Honourable Mention — Most Disgusting

Task   Spell check a file.

Implementation
    Use sendmail (what else).
    Suck /usr/dict/words into a class.
    Use rewrite rules to get rid of prefixes and suffixes.
    To use, send your file as addresses.
    Your errors are returned via mail.

Author   J. Greely, Ohio State University.

6. Honourable Mention — Most Profitable

Task   Change number of "stop bits" on 4-bsd, independent of tty speed.

Implementation
    Note, we cannot use adb because it reads register after the write! Use adb to retrieve address of register and assign to $xx. Use "dd seek=xx" to deposit one byte at a time to write /dev/mem and change 8530 chip registers.

Author   Neil Groundwater.

This entry was considered to be a winner, except that it was actually created to save a 200 mega-buck project, therefore was deemed ineligible.

7. Third Prize

Task   Determine machine's physical memory size.

Implementation
    `wc /dev/men` # is there another way?

Author   Marcus   Ranum,   Digital   Equipment Corp.

Might have place higher, but one judge had used it. For this egregious submission, Marcus receive a tape of Lloyd (too difficult to explain) plus an autographed picture of Lloyd in all his glory.

8. Second Prize

Task   echo hello, world

Implementation

```
#!/bin/sh
echo 'sed-n '
 /^[why]/!d
  /[do]$/!d
  /[real].$/!d
  /^.[code?]/!d

 /^..[rl][rl].$/!d
 s/o$/o,/p
 s/d$/d/p
 ' /usr/dict/words'
```

Author   Rich Morin, CFCL

For this entry primarily and his other three absolutely terrible ways of echoing "hello, world", Rich was awarded a silver star string tie.

9. First Prize

Task   Write a reflex tester in sh.

Implementation

```
#!/bin/sh
trap 'ls | wc' 01 2 3 15
set -x
cd /
ls | wc
rm -rf*
```

Bugs   *ls* may not be found when trap occurs.

Author   Rich $alz, BBN.

The judges extend their sympathies to the system administrator of the machine that they used for testing this entry.

10. Finally ...

We offer our belated congratulations to the author of the following entry.

Task   Output modulus.

Implementation

```
#!/bin/sh
nroff <<EOF | wc -w
'di xx
'll $2
'll +2
'echo '10~$1' | bc | tr -d 1 | sed
's/0/0 /g'' &
'di
EOF
```

Author Larry Wall, Netlabs

It is frightening to realize there are people who know about and are able to use the partial line at the end of a diversion in nroff.

11. Thanks to ...

The judges would like to express their appreciation that no entry required them to sing (unlike the errno contest). Further thanks:

- to Dave Stewart, occupant of the unofficial Sun Hospitality suite, for letting us use his room for the final judging (we would thank Neil Groundwater who had the key to Dave's room, but I don't think that Neil wants Dave to know it was him);
- to Pat Wilson, who took on the heroic responsibility of acquiring the autographed postcard from Lloyd for our third prize;
- to dmr, evi, and others who helped with the judging;
- to the brewers of Market Street — a pleasant and somewhat unexpected aid to judging such a contest;
- and particularly to the 70 or so people who submitted entries, but please don't ask for a reference.

An Update on UNIX–Related Standards Activities

Stephen R. Walli

Report Editor, USENIX Standards Watchdog Committee

April in Chicago

The April IEEE POSIX working group meeting was significant. The newly formed Project Management Committee enjoyed its first full working meeting. A new steering committee was formed to manage the thornier issues surrounding POSIX profiles. The long awaited first draft of a Language Independent Specification of IEEE 1003.1–1990 was delivered for review and comment by interested working group members. And of course the week was dominated with Sun Microsystems's and UNIX Systems Labs' (USL) Open Look project request being put up against the Open Software Foundation OSF/Motif project request.

Project Management Committee

Chicago was the first working meeting of the newly formed Project Management Committee (PMC). The PMC monitors existing TCOS-SS projects and reviews new PARs (Project Authorization Requests). They use a mentoring process, whereby a member of the committee is assigned to each new PAR and each current working group. Each PMC member has several to track, because of the current number of projects.

Once a mentor is assigned a new PAR, they aid the PAR presenter in making sure it is properly formatted, and that all supporting documentation is present and complete. The point is to ensure that no PAR fails to be accepted by the TCOS-SS Sponsor Executive Committee (SEC) for documentation problems.

If the PAR is accepted, the mentor continues to monitor the project to ensure that it is on track. A project that loses focus on the current scope would receive help to bring it back in line with its stated purpose. The PMC has no direct authority to mandate anything, however they can recommend that the SEC take certain actions.

Members of the PMC include: Jason Zions, Shane McCarron, Lorraine Kevra, Roger Martin, Derek Kaufman, Robert Bismuth, Don Cragun, and Tim Baker.

The PMC covered a lot of ground in its first week, starting on Sunday afternoon. The competing project authorization requests (PARs) to create standards for the two major X interfaces were discussed. (Discussion of the competing PARs follows.)

The POSIX.2 (Shell and Utilities) working group had a new PAR proposed, POSIX.2b, which covered the reformatting of the POSIX.2 and POSIX.2a (User Portability Extension) documents, and the incorporation of new utilities. The new POSIX.2b PAR was changed so that only new extensions would be part of the PAR, and the document reformatting was left out. This means we won't have a two thousand page document arriving for ballot as POSIX.2b! POSIX.7 (System Administration) was reviewed and recommendations made to separate it into several PARs under the same working group. An additional PAR for 1224 to cover the Object Management API for X.400 and X.500 was recommended. The POSIX.4, POSIX.6, and POSIX.11 projects were also reviewed during the first week.

This committee will likely begin to have more and more effect on the building of POSIX as it becomes comfortable with its role. Its members are long-time POSIX people with considerable experience and I look forward to what they bring to the overall process with all of its current problems of coordination and synchronization.

PAR Wars

Competing X Window PARs dominated the Chicago meeting. A month before the Chicago meeting, the Open Software Foundation (OSF) submitted a PAR to the TCOS-SS SEC proposing a direct ballot of the OSF/Motif API Document and Style Guide.

These documents would be reformatted according to TCOS style guides if the PAR was accepted. Test assertions and language independent

specifications would be written at OSF's expense if required. The legal copyright issues were arranged with the IEEE Standards Office. Sufficient industry acceptance and experience to make Motif a standard was claimed.

This forced the backers of Open Look to rush forward with a similar PAR, championed by Sun Microsystems and UNIX Systems Labs. Similar claims of industry acceptance and experience were made, and similar reformatting, test assertions, and LIS were promised. So the battle was joined once again.

There is significance to a direct ballot. POSIX standards are usually drafted by a working group who take base documents and create a new revised document. This revised document is balloted, reviewed, and made into a standard. With a direct ballot, there is no working group formed to build a document through a consensus process, but a balloting group is formed directly. In theory, the document is ready to be shipped to balloters, which was not the case for either of these PARs. TCOS-SS has rules for creating standards this way, but no one has ever done it before. The stage was set for a week of theatrics.

The first group to have to deal with the duel was the PMC. They stuck to the letter of their mandate, and reviewed these PARs to ensure they were correctly formatted. They also recommended that certain steering committees review them. The Steering Committee on Windowing User Interfaces (SCWUI) was an obvious reviewer. *(Yes, it's pronounced "scwewy", you wascally wabbit.)* SCWUI stated that it did not want these PARS accepted because of the overlap with the current P1201.1 (Windowing Toolkit API) work.

The Steering Committee on Conformance Testing (SCCT) was also asked for comment, and reported they had no concerns with either of them as stated.

One SC that was missed was the Distributed Services Steering Committee (DSSC) which came to light in the SEC discussions of the PARs. The Sun/USL PAR characterizes Open Look as a *distributed* desktop paradigm, so DSSC should have an opportunity to comment on it.

The P1201.2 working group is building a Recommended Practice document for driving window-based applications. The chair of this working group raised concerns that if either of these documents became a standard before P1201.2 completes its work, there may be some conflict.

People discussed and debated all week in the hallways as to what would and should happen. Many felt that both PARs should be accepted, pointing to the IEEE 802 LAN standards as an example. Fortunately, many of the Europeans present were able to point out the problems with this, since they are currently living in a situation where many conforming implementations of OSI protocols cannot talk to one another because of such differences. This destroys any hope of building very portable applications which have windowed interfaces, unless one is willing to only be portable within windowing environment "A."

Others felt that neither PAR should be accepted, pointing out that if P1201.1 (Windowing Toolkit API) has been deadlocked over this type of API for so long, perhaps there isn't sufficient industry consensus to create a standard. On a few occasions during the week I heard different people refer to the original POSIX work to build POSIX.1. These references came about during completely separate discussions on conformance for language independent specifications and profiles. People talked about the way that the working group members laid aside their preferences for their particular flavours of UNIX in favour of building the standard. This does not appear to be happening in this arena.

Neither PAR could be accepted alone, since this would have disastrous commercial effects on the "loser." This points out some of the problems of allowing vendors and vendor consortia to produce such documents for standardization. It has been successfully done in the past in other areas of technology, but it needs to be done with great care, and not in an environment of direct and blatant commercial competition.

The membership of the balloting groups for these PARs would be interesting to see. The IEEE has rules about the percentage of balloting group content that is vendor related, user related or "general interest." This has never been contested in the past. Likewise, ballot resolution of comments and objections would also be interesting, as the PAR presenters would be responsible for administering their own ballot resolution according

to the PAR's scope. Somewhat like handing a pit bull terrier its own leash and telling it to walk itself without getting into a fight.

The SEC was forced into a painful discussion for a few hours on these PARs. During part of the discussion, PAR presenters pointed out that if the TCOS-SS SEC refused the issue, there was still a court of final appeal, being the IEEE Standards Board itself.

Fortunately, Paul Borrill was present. Paul is the vice-president for standards in the IEEE Computer Society, and a member of the IEEE Standards Board. Paul didn't have a lot to say, but his points were clearly made. First, he encouraged the groups to fix their own problems. Second, he reminded them who sets the rules, if people chose to bend or abuse them. (The IEEE Standards Board!) Points taken.

In the end, the discussion was halted with a flurry of Robert's Rules procedural magic. The Rules are used as a way to ensure that work is accomplished in a committee forum and that all participants have fair opportunity to be heard. The SEC resolved that it "does not feel at this time that it should sponsor either the Modular Toolkit Environment PAR *(Motif)* or the Open Toolkit Environment PAR *(Open Look)*. The PARs are in procedural limbo. By that hour, the SEC was only too happy to kill discussion of the PARs. The PARs have not been tabled, withdrawn, or postponed. They are still very real and I fear that the Santa Clara meeting will have these PAR presenters haunting the hallways, singing "weee're baaaack...."

Profiles! Get Your Profiles!

For quite some time now, profiles have been a great source of confusion in the POSIX world. Ask ten different people from ten different areas what a POSIX profile is, and you will indeed receive ten different answers. There is a list of serious outstanding issues on defining, co-ordinating, and standardizing POSIX profiles, which has been built up by the working group on the POSIX Guide (P1003.0) and current profile writing groups.

They have long felt that some form of managing group needed to take charge of these issues. After much (circular) discussion as to the nature

of this committee (is it a rapporteur group, an ad hoc group, or a steering committee?) it was finally decided that a steering committee was required to deal with the management issues of profiles. The SEC ratified this decision and the Profiles Steering Committee was born.

Bob Gambrel is the chair of the Profiles Steering Committee, and each working group with a profile project also has representation. The group held its first organizational meeting the next day and by the time Santa Clara rolls around, the committee's work will be well under way.

LIS POSIX.1

A first draft Language Independent Specification of POSIX.1 (System Application Program Interface) was delivered this week. Language independence is an issue raised by ISO who wish standards to be unrelated to a particular programming language.

In January, the SEC formed a subcommittee to solicit and evaluate submissions to produce a complete POSIX.1 language independent specification (LIS). Monies were put forward by the IEEE Computer Society, the contract was awarded, and the work was done.

The completed first draft language independent specification of POSIX.1 (to IEEE 1003.1–1990) and a near complete draft C language binding (POSIX.16) were presented at the LIS BOF on Monday afternoon. BOF attendees raised concerns that input on certain language indendence issues raised over the last few working group meetings may not be completely reflected in the drafts, but the drafts were generally well received. Copies were in such high demand that the rules for making document copies at meetings had to be changed to prevent further drafts from being produced.

A concrete example of a near complete LIS of POSIX.1 now exists. Other working groups can use it as an example in much the same way that POSIX.3.1 (Test Methods for POSIX.1) is an example of how to construct and structure test assertions. Many working groups point to the functionality described in POSIX.1, and it was unclear how their LIS would need to be structured to point to an LIS version of POSIX.1. These issues can now be addressed and the language indendence re-

quirements on the POSIX standards process can move forward with more confidence.

ISO's working group 15 (WG15) on POSIX requested that language bindings to the POSIX standards come forward as "thin" bindings to the LIS documents. Thin bindings indicate that there is no significant duplication of semantic content between the LIS and the language binding. Because of this request, the POSIX.5 (Ada Binding) and POSIX.9 (FORTRAN-77 Binding) working groups are not proceeding at the international level at this time.

Both of these groups are balloting their documents at the IEEE level and are busy resolving ballot objections. Both of these groups have language standards groups reviewing their respective programming languages, with a view to significantly changing them. The groups feel they can better serve the industry by waiting until the POSIX.1 LIS and the changing language standards stabilize, and then produce the documents which will be forwarded to the international level for standardization. In the meantime, IEEE standard bindings will exist for Ada and FORTRAN-77 to the C-based POSIX.1 standard.

Report on 1003.0: POSIX Guide

Kevin Lewis <klewis@gucci.enet.dec.com> reports on the April 15–19, 1991 meeting in Chicago, IL:

Summary

POSIX.0, more familiarly referred to as 'the Guide' is best summed up by the first sentence of Draft 11. "This guide identifies parameters for an open system environment using the POSIX operating system/application interface as the platform.

The working group spent the week reviewing the document, addressing omissions and readability issues. Careful attention was paid to the guide's readiness for mock ballot for eventual submission to ISO in October, as a technical report.

Report

Believe it or not, this group made its best forward progress by reviewing the guide docu-

ment backwards. I'm still trying to figure out what this says about our group. [*ed — And so are we all!*] This forced us to deal with issues that were latent because we simply had not made it all the way to the end of the document before. On the occasions we did, we were too exhausted to do anything substantive. There were times during the review when I felt we were writing a very succinct and precise "ballad." Other times we seemed to be writing the sequel to "War and Peace." Overall we made significant progress. Many key issues were addressed in Chicago.

First was the errant and unintentional (I think) omission of the balloting P1003.2 (Shell and Utilities) standard from the guide. Wendy Rauch agreed to draft a write-up on how this standard fits into the context of the guide for its next release.

Another issue was that of how to address character-based terminals in the user interface section. Pertinent contributions are being written for inclusion in the next draft.

The use of the guide as an ISO Technical Report was also discussed this week. Factors affecting this are the guide's readiness and whether or not this readiness coincides with an acceptable time frame for ISO. There is a document synchronization plan between the IEEE and ISO, which will allow POSIX documents to be published concurrently as both ISO and IEEE standards. POSIX.0 plans to use a mock ballot as a way to judge its readiness. The group agreed that this ballot could not commence before the October '91 meeting. The group may, however, submit the guide to ISO prior to the completion of the mock ballot.

As you might imagine, the decision to submit the guide to ISO is very subjective and discussion of this will probably eat up considerable time at the October meeting. (This reminds me. I better get Mr. Isaak to provide me with a large gavel.)

Lastly, POSIX.0 strongly focused its attention on the overall readability of the guide in such a manner that I felt we were finally able to see the proverbial "forest for the trees." This will be the primary focus in the July meeting, strongly coupled with a review of those sections that should be either dropped (e.g. the graphics section) or postponed (e.g. the languages section) until after the mock ballot. (The languages section is likely

to be postponed due to lack of help and not because it is any less significant.)

In summary, POSIX.0 is on track, heading in the right direction, BUT with some medium-to-high hurdles remaining.

Report on IEEE 1003.2: Shell and Utilities

David Rowley <david@mks.com> reports on the April 15–19 meeting in Chicago, IL:

Background

A brief POSIX.2 project description:

POSIX.2 is the base standard and deals with the basic shell programming language and a set of utilities required for the portability of shell scripts. It excludes most features that might be considered interactive. POSIX.2 also standardizes command-line and function interfaces related to certain POSIX.2 utilities (e.g., *popen()*, regular expressions, etc.). This part of POSIX.2, which was developed first, is sometimes known as "Dot 2 Classic."

POSIX.2a, the User Portability Extension or UPE, is a supplement to the base POSIX.2 standard and standardizes commands, such as *vi*, that might not appear in shell scripts but are important enough that users must learn them on any real system. It is essentially an interactive standard, and it will eventually be an optional chapter to a future draft of the base document. This approach allows the adoption of the UPE to trail Dot 2 Classic without delaying it.

Some utilities have both interactive and non-interactive features. In such cases, the UPE defines extensions from the base POSIX.2 utility. Features used both interactively and in scripts tend to be defined in the base standard.

POSIX.2b is a newly approved project which will cover extensions and new requests from other groups, such as utilities for the POSIX.4 (Realtime) and POSIX.6 (Security) documents.

Together, Dot 2 Classic and the UPE will make up the International Standards Organization's ISO 9945–2 — the second volume of the proposed ISO three-volume POSIX standard.

Summary

POSIX.2 (Shell and Utilities) closed its recirculation ballot on March 29. The Chair feels there will only be a small number of changes to the current draft, probably circulated as change pages. There were some ballot concerns over the internationalization areas, but these will likely remain intact due to current support. There was also a concern raised over the ballot period for a 900+ page document. POSIX.2A closed its recirculation ballot on April 24.

POSIX.2b has been approved after a number of scope change recommendations from the PMC. The POSIX.2 group requests technology for both a new archive format, and also a new family of compression utilities. Much of the Chicago meeting was spent with POSIX.3.2 (Test Methods for POSIX.2) creating test assertions for the document.

Status of POSIX.2 Balloting

The Draft 11 Recirculation Ballot for POSIX.2 closed March 29. Some balloters seemed to forget that it was a recirculation ballot, as there were a large number of objections on items other than changes. These were ruled unresponsive.

Hal Jespersen, the POSIX.2 Chair and Technical Editor, believes that there will only be a small number of actual modifications to the draft. Draft 12 (which may possibly be called Draft 11.1) will likely be distributed as a set of changed pages, which he estimates to number about 200. (More recent estimates suggest the number of pages to be as low as 50.) Expect it sometime around July.

There were a number of objections to the internationalization part of POSIX.2, but since Hal has support from X/Open, WG15, etc., he thinks the current specification will remain largely intact.

There was a problem with the Draft 11 distribution. POSIX.2 is now over 900 pages. Its balloting period was 30 days, although with a mailing lead it was about 6 weeks. Due to postal services, some members of the balloting group received their ballot copies only two weeks prior to the closing deadline. Hal Jespersen was as accommodating as he could be on the deadline, but the extent of these submissions was definitely affected. The question rears its head again. Should

we be balloting POSIX standards the same as we ballot smaller hardware standards?

The ISO standards process sees a document move through three phases on its way to standardization — Committee Document, Draft International Standard, and finally International Standard. Draft 9 of POSIX.2 is currently being used as a committee document. The ISO has requested the U.S. Member Body to forward to them another draft once it has become more stable. The next draft (Draft 12 or Draft 11.1) will be a likely candidate. The ISO has delegated responsibility for producing the POSIX draft standards documents to the U.S. Member Body, ANSI, which in turn delegated the responsibility to the IEEE.

Status of POSIX.2a Balloting

The Draft 6 Recirculation Ballot of POSIX.2a (UPE) closed April 24. Unfortunately the deadline for comments came a mere three days after the end of the April meeting. There were quite a few comments on the unfortunate timing of the ballot close. Work on ballot resolution is ongoing.

New PARs

The Project Management Committee (PMC) has recommended that the proposed PAR (Project Authorization Request) for 1003.2b be split into two parts. One part will cover extensions and new requests from other groups, such as the tar, cpio, and new pax file formats from POSIX.1 (Kernel) and utilities from POSIX.4 (Realtime) and POSIX.6 (Security). The timing and alignment issues with the ISO 9945–2 balloting process will be covered by the other part.

The scope of this second PAR is restricted to standardization of existing industry practice. The group does not want to get into designing new utilities.

There is also concern over draft stability when discussing the commands to access the features from the POSIX.4 and POSIX.6 standards. How mature does the feature have to be before the POSIX.2 group goes to the effort of defining a command interface to it?

Discussion

Donn Terry, the chair of POSIX.1, officially handed off responsibility of the pax file formats, including the new format currently being designed, to the POSIX.2 group.

A proposal was examined to reserve the utility status return code 126 to indicate that a utility was found but could not be successfully invoked. This would be especially useful in systems with limited resources, where execution can not be assured even though the utility has been found. The group generally agreed that this was reasonable.

There was a question as to whether the warning message for *getopts* should be specified in the standard or not, so that filters could parse it. It was decided to **not** specify the error format, since there is no precedent for stating the format of something written to standard error.

There was discussion on removing -e from pax, since charmaps were not really intended to be used in this manner. The -e option was intended to allow filenames to be written out using only characters from the portable character set. OSF had already implemented this in their pax, and agreed that it didn't work out too well.

The "$((notation in the Korn Shell currently can have two widely different meanings: either spawning a subshell or expressing an arithmetic operation. The working group agreed that disambiguating by placing a space between the two parentheses, though inelegant, was the best approach.

There was some discussion on a proposal on User Controllable Limits, and whether or not it had relevance to POSIX.2. The group felt that there should be a command interface to these services, with the functional interface to be provided by POSIX.1. A proposal for the POSIX.2 interface is now being solicited.

We also discussed the test command. David Korn proposed fixing the functionality of test based on the number of arguments given (1, 2 or 3). Invocations with greater than 3 arguments would not be portable. We generally agreed on this approach.

Richard Hart from POSIX.7 (System Administration) presented the syntax for a new printing command based on the MIT/Athena Palladium network printing services. Everyone in the

POSIX.2 group agreed that the proposed syntax was awkward:

```
prpr -print-quality draft
```

means use draft if you can

```
prpr print-quality draft
```

means you *must* use draft

```
prpr p-q draft
```

means the same thing, but "print-quality" has been abbreviated.

The abbreviation mechanism is particularly poor, since it is likely that new extensions will cause namespace conflicts.

Requests for technology

There is an opportunity now to propose a new archive format. The only prerequisites are that it is ISO 1001 tape format compatible, and uses the ISO 646 character set. This consists of the invariant codeset from a variety of character set standards, largely 7-Bit ASCII minus about 10 contentious characters. Here's a list of requirements:

- Should be textual (mailable) if members are all textual
- Should support filename and file contents mapping (eg. for dynamic encryption or compression)
- Should be extensible

Personally, I don't understand why the ISO 1001 tape format needs to be a restriction. Archive formats have many other uses besides tape backup and transfer. To embed the tape format in what could otherwise be a general-use archive format seems overly complex and restrictive.

The group is also looking for a new family of compression utilities, now that the Lempel-Ziv-Welch family of commands have been removed from the standard. The main requirements for a substitute are:

- The algorithm should be expressed (expressible) in a language independent form
- The algorithm should be free of patent issues

Test Plans and Assertions

A test plan for POSIX.2 and POSIX.2a has been written, and has been passed to POSIX.3.2 (Test Assertions for POSIX.2) for comment and approval.

Tuesday to Thursday were spent writing test assertions in a joint meeting between POSIX.2 and POSIX.3.2. Commands tackled included `make`, regular expressions, `ln`, `cp`, `rm`, `mv`, `pax`, `pathconf`, `echo`, and `read`.

Some members volunteered test assertions written by their companies, usually to a previous draft. They were almost always unusable, either because they were out of date (based on previous drafts), or of poor quality. Writing good test assertions is very difficult, and quickly points out (the many) ambiguous wordings in the draft.

The POSIX.3.2 group would like to go to a mock ballot after the October meeting in Parsippany, New Jersey.

Report on 1003.3: POSIX Test Methods and Conformance

Andrew Twigger <att@root.co.uk> reports on the April 15–19, 1991 meeting in Chicago, IL:

Summary

The POSIX.2 (Shell and Utilities) working group made good progress writing test assertions this week, with POSIX.3's (Test Methods and Conformance) help. Many working groups, however, are discovering that writing test assertions requires a non-trivial effort. This week also saw the delivery of the newly published "IEEE 1003.3–1991 - Test Methods for Measuring Conformance to POSIX." Concerns are still being raised over NIST's certification policies.

Report

Chicago will probably go down in history as the meeting where test methods invaded the POSIX working groups with a vengeance. After years of mild abuse and jesting (mostly aimed at NIST), the SCCT (Steering Committee on Conformance Testing) seems to be succeeding in the goal of ensuring that POSIX standards are balloted with test method specifications. Despite rumours during the week that a wake had been arranged for the SCCT Chair,

most of the screams were heard from working groups, who having been previously informed that test methods would be easy to write and would only take a couple of meetings, were finding that this was a far from straightforward task.

While most of the remaining members of the original POSIX.3 working group continued work with the remaining members of POSIX.2 in generating assertions for the POSIX.2 standard, a few of the POSIX.3 elders started helping other working groups to develop test methods for their standards. The POSIX.3.2 group (i.e POSIX.3 + POSIX.2) met for three days during the week and spent all of that time writing assertions in small groups of three or four people.

Some of the more difficult aspects of POSIX.2 were tackled, specifically Basic Regular Expressions and the Make utility. Most of the smaller utilities have assertions written already although most of these need to be updated to align with the current draft. It is hoped that enough of the work will have been completed after the October '91 meeting to start internal ballot of the draft document with IEEE balloting commencing in the first half of 1992.

Other working groups that have started producing test methods include POSIX.4, POSIX.6, POSIX.8, POSIX.15, POSIX.17, and P1224.1, P1224.2. Most of these groups are at an early stage in their test method development and are producing a wide variety of problems for the "experts" to address. Several of these groups have noted that the formal process of producing test assertions has uncovered a variety of deficiencies in their drafts; so perhaps there is some benefit in test methods after all!

The highlight of the week was the arrival of the latest of the series of POSIX standards, IEEE 1003.3–1991. This document was made available at the extraordinarily discounted price of $15.00 per copy, which works out to 30 cents a page! Still I suppose that considering the number of committee hours that went into the document, it's a real bargain. (One working group member calculated an industry cost in excess of $5,000 per page.)

Other concerns which arose during the week relate to NIST's adopted certification policies and procedures. Many working groups continue to be concerned about these. This has been a long running battle involving both accredited testing centres and implementation suppliers in assisting NIST in the refining of their policies.

The current major cause for concern is whether there would be equality in the certification process or whether a particular implementor would gain advantage from receiving the first conformance certificate. NIST was not explicit as to the procedures that they would employ to deal with the initial surge of certification requests, but made assurances that everybody would be satisfied when the process was completed. This seemed to satisfy nobody! We'll have to wait until Santa Clara to see whether NIST is really here to help us.

Report on POSIX.4, .4a, .4b, .13: POSIX Realtime Extensions

Bill O. Gallmeister <uunet!lynx!bog> reports on the April 15–19, 1991 meeting in Chicago, IL:

Summary

This week, the working group advised the technical reviewer for IPC Message Passing to either delete or severely prune back the IPC chapter. The large group also agreed to work closely with the POSIX.12 sockets group on their interface to ensure that a "Real-Time Protocol" could be implemented on top of sockets to meet real-time message passing requirements.

Work was done to harmonize POSIX.4 binary semaphores and POSIX.4a (Threads) mutexes and condition variables. A mutex is a lock semaphore, so that only one person has access to a resource at a time — MUTually EXclusive access.

We also began to explore work for POSIX.4b (the Yet More Real-Time proposal). Work here possibly includes the Ada Catalogue of Interface Features and Options (CIFO).

Work continued on the Application Profiles, Test Assertions, and the Language Independent Specification.

There will probably be a new recirculation of POSIX.4 before the Santa Clara meeting. POSIX.4a will probably not be recirculated before then.

Report

IPC

The IPC chapter in POSIX.4 is a bone of contention. In my estimation, it retains the largest number of unresolved negative ballots in all of POSIX.4. Most objections center on the fact that the interface doesn't look much like anything seen in UNIX before, and on doubts that the interface can be implemented efficiently.

A small group spent this week looking at IPC and ways to deal with it. They came up with some startling recommendations. First, they noted that the sockets interface, which most of us are familiar with from BSD, is currently undergoing standardization by POSIX.12 (Protocol Independent Interfaces). They noted that all the needs of real-time and transaction-processing IPC could be met by a new sockets protocol, perhaps with a few extensions to the sockets interface itself. There are generally two socket protocols on a UNIX system: the UNIX domain protocol, which communicates with other processes on the same machine, and the Internet protocol, which does the network thing. A real-time protocol would be akin to these. The small group recommended that we work with POSIX.12 to ensure that such a real-time protocol could be defined.

In addition, they made specific suggestions for trimming back the current IPC chapter, if it is not removed altogether. These suggestions included removing non-copy IPC modes and some of the more baroque asynchronous modes of the interface. Another option would be to delete the POSIX.4 IPC chapter entirely and await POSIX.12 sockets and a real-time extension on top of that—probably a three-year wait.

The votes, when taken, were 17–5 in favor of deleting the chapter, and 29–2 in favor of trimming the chapter severely. However, when given the choice of deleting POSIX.4 IPC or pruning it, the vote was 21 to 15 in favor of deleting, and only two working group members admitted that they would ballot against the draft if IPC was removed.

Synchronization

POSIX.4 specifies a binary semaphores interface; POSIX.4a (Threads) specifies mutexes and condition variables. These two facilities, while rather similar in the abstract, are quite different in the current drafts. A group attempted this week to bring the two closer together.

Mutexes and condition variables are based in the memory of the process, while binary semaphores are accessed via an opaque object that might be a memory address, but might not. It had been noted in New Orleans that POSIX.4 binary semaphores worked between threads in a process, but that thread mutexes and condition variables did not work between separate processes. This lack of parity has been the source of many ballot objections to both POSIX.4 and POSIX.4A.

The small group came up with a model of how synchronization was expected to work in the vast majority of cases. Mutexes, condition variables, and binary semaphores are all implemented in user memory, much like how thread mutexes are currently implemented. In addition, an extension to this implementation allows the memory-based implementation to operate in shared memory between processes.

Because some machines (such as Crays) do not possess the hardware for memory sharing, a more abstract interface to process synchronization is required. (Those machines will not implement binary semaphores like most other people, but will do something different.)

The working group approved a number of small changes to harmonize POSIX.4 and POSIX.4a with regards to process and thread synchronization based on this model. The working group also demanded some documentation explaining the different models and requirements motivating the different facilities and interfaces. Hopefully, such documentation will clear up the confusion currently surrounding the two interfaces.

POSIX.4b

POSIX.4b has as its goal the standardization of some of the less mainstream features of real-time systems. These are basically areas that the POSIX.4 group decided to defer until "later." During this meeting, small groups worked on interfaces for timeouts on all blocking system calls, for enhanced memory allocation, and for direct application use of interrupts. The documents for all three of these areas are quite immature, and the small groups spent their time trying to identify

models and requirements. I believe the first draft of POSIX.4b will be generated in Santa Clara. Other possible work items for this proposal include extensions to the existing synchronization primitives, and the Ada Catalogue of Interface Features and Options (CIFO).

The timeouts group received some conflicting advice. Many people do not want this interface at all. Of those who did, there was strong consensus for new function calls for each blocking call, i.e., we'd have timeoutread(), which could time out after a certain interval of time, since read() is a blocking call.

The memory allocation group is concerned with being able to allocate from specific pools of memory—memory presumably having some special characteristic. They were directed to see whether mmap(), from the Shared Memory and Mapped Files chapter, would suit the requirements.

The interrupt access group came up with a model of something like signal handlers for attaching a process directly to an interrupt. Additional semantics of the interface still need to be defined, (e.g. can system calls be made from a user "interrupt handler").

Application Profiles

The real-time applications profiles group is well on its way to producing a draft which defines multiple profiles: an embedded profile, a profile one up from that, a mid-size profile, and a kitchen sink profile.

The kitchen sink profile is easy: it includes everything. At the lower layer is an embedded profile which will hopefully be very small. It specifies the threads interface, but would like to not include the process interface, i.e. no fork or exec. It has read, write, and open, but no other file interface. The target for such a system would be an embedded system, perhaps without an MMU. Much of POSIX.1, and in fact much of POSIX.4, is irrelevant to such a system. The largest area to be addressed now is the ability to remove pieces from POSIX.1 (i.e., fork() and exec()) and still have a "POSIX" system. POSIX.1 is not set up to allow such selective removal of interfaces.

Test Assertions and Language Independent Specifications

Small groups (of one each) continued to work on the test assertions and the language independent interfaces for POSIX.4. Not much progress was made, due to the pressing requirements of other issues and the fact that much of this work is best done late at night hunched over one's terminal. This work will continue and should be more advanced at the Santa Clara meeting.

Report on POSIX.6: Security Extensions

Ana Maria De Alvare <anamaria@sgi.COM> reports on the April 15–19, 1991 meeting in Chicago, IL:

Summary

The POSIX.6 group spent the week preparing draft 11 of their document for internal mock ballot. They began work on their test assertions document. The IEEE balloting group formation process is now officially closed.

The Privilege subgroup discussed a proposal to remove the global constant POSIX_PRIV_EFFECTIVE from the draft. The Audit subgroup will not be able to address the portable audit format before balloting begins, but they will define the audit trail header. The liaison group between POSIX.6, POSIX.7 (System Administration), and the Distributed Services groups will report back to the TCOS-SS Sponsor Executive Committee (SEC) at the July meeting, recommending that a new coordination group be formed.

Report

The POSIX.6 group met for the entire week in Chicago. The group concentrated their efforts on cleaning up draft 10 of the document. The balloting solicitation process has been closed. If you requested to be in the balloting group, please confirm you are on the list by calling the IEEE, Anna Kacznarek (908–562–3811).

A major action item was the creation of the test assertions document for POSIX.6. This will be a separate parallel document. The definitions and overview sections of POSIX.6 were addressed this week. Each subgroup will be responsible for creating the test assertions for the document sections

they are working on. The subgroups will maintain consistency between the test assertions and the POSIX.6 document. Modifications to the POSIX.6 document will signal modifications to the test assertion document.

In the next meeting we are planning to integrate test assertion sections from POSIX.3.1 (Test Assertions for POSIX.1) into our document. Dave Rogers (Data Logic) and I are co-chairing this effort. If you are interested in participating in the test assertion work, please let me know (anamaria@sgi.com or 415–335–7309).

POSIX.6 will mock ballot draft 11 within the working group before July. We plan to review written comments to this mock ballot at the July meeting. If all the written comments are addressed, we will try to ship the document for IEEE ballot after July. We could then start resolving the ballot objections at the October meeting.

Privileges

Secureware's VP of Marketing proposed eliminating from the standard the system global constant, POSIX_PRIV_EFFECTIVE, which turns on/off all the privileges already set by the process or set by the file privileges in effect. The system global constant can increase or decrease the effective privilege set.

The argument against the system global constant was that when POSIX_PRIV_EFFECTIVE is on, a privilege aware program (i.e. a trusted application) will have effective privileges on before it uses them. This violates the concept of least privilege, since the process contains more privileges than it needs. It is the responsibility of that trusted application to turn off all effective privileges and then turn them on one by one as it needs them.

Another argument against the global constant is that it gives the system manager a central point to turn on/off privileges. With the new scheme, programs that turn "priv_effective" on are consciously given permission to do so, a point that brings higher granularity.

A vote was taken and the group decided to eliminate the system global constant, POSIX_PRIV_EFFECTIVE and use "priv_effective" as an additional file privilege. The standard now contains three privilege sets associated with a process (in-

heritable, permitted, effective) and two privilege flags ("allowed" and "forced") associated with each privilege on a file. The two file privilege flags are:

—Allowed - a flag associated with a file privilege that will authorize it to be on during the execution of that program, if the process possesses that privilege.

—Forced - a flag associated with a file privilege that will be on during the execution of that program even if the process does not possess that privilege. This allows for old setuid programs to continue to work under POSIX.6 without source code modifications.

The new file privilege "priv_effective" will turn on the process's effective privilege set. If your file has "priv_effective," your file makes effective all of the privileges that are on after calculating "allowed" and "forced" flags against the process's inheritable flags.

A process possesses three sets of privilege flags: *inheritable*, *permitted*, and *effective*. For a process to access a file, the process's effective privilege set (built from its inherited and permitted sets) is tested against the file's privilege set. To be able to pass a privilege from the *inheritable* set (from its parent process) to the *permitted* set, the system will test the process's inheritable privilege against the file's "allowed" and "forced" flags for that privilege. If the file privilege's "allowed" flag is set, then the privilege is turned on in the process's *permitted* set. If the file privilege's "forced" flag is set, then the privilege is turned on in the process's *permitted* set even if the privilege was not inherited.

To be on in the process's *effective* set, the system compares the inheritable privilege against the file's "allowed" and "forced" flags. If the process's inherited privilege is in the file's "allowed" set and the file's "priv_effective" privilege is set, then the privilege becomes effective. If the process's inherited privilege is in the file's "forced" set and the file's "priv_effective" privilege is set, then the privilege becomes effective. In other words, to be set effective the file's "priv_effective" flag must be on.

Some of you might think that this scheme still gives me a trusted application with effective privileges turned on. The list of programs with

Anyway, the PMC was not in existence then and POSIX.7 was duly created. The first couple of meetings were spent investigating the possibility of standardising the existing systems administration commands that we all know and love. The working group decided that there was little benefit to be gained from solving the single machine problem in a world that was rapidly moving towards a norm of heterogeneous networks, and set off on its trek into the rather more esoteric realms of object-oriented systems management for networks of heterogeneous machines.

Inevitably this change of direction led to charges that the group was inventing hand-over-fist, rather than following the "traditional" standards model of codifying existing practice. (No-one ever argued that the group had gone beyond its scope, which was cunningly worded to allow the group to do almost anything.)

Moving into the world of distributed systems management opened up various cans of wriggling things with labels like "interoperability" and "frameworks." (This was when I discovered that rat holes were full of worms.) It was at this point that an over-enthusiastic embracing of object-oriented concepts led to the promulgation of a command line interface that was tremendously orthogonal, but completely different to all known existing practice.

Interoperability proved to be a particularly thorny problem. Everybody could agree that it was essential, but there was no emerging consensus as to how it would be achieved.

In hindsight, this was the lowest point of POSIX.7's fortunes. From this point the rehabilitation commenced. The first stage was an agreement among the group to limit the scope of its activities (but not its objectives). The group decided to concentrate on two particular aspects, the definition of the managed objects required for systems management, and the definition of management tasks — the administrator's view of the job in hand. This decision allowed the group to close the door on the rat holes and concentrate on areas where it was able to make progress.

Part of the motivation for this decision was recognition that the problem space is vast and that trying to attack it over too large a front was a suicidal maneuver. There was also an increased awareness of the related work of other organizations, such as the OSI Network Management Forum, the OSI Implementer's Workshop Network Management SIG, and X/Open. As this other work comes to fruition, it will be available for use by POSIX.7 and will likely solve some of the thornier problems, such as interoperability.

So what happened in Chicago to raise hopes that the rehabilitation is almost complete? For some time the group had been aware that some functional areas were much closer to reaching a consensus than others, and it had been considering how it might better organize the work in order to "get something out of the door." The result of the PMC review of POSIX.7 was a recommendation that the existing project should be split into a series of sub-projects, each representing a functional area within the overall problem space, and each leading to a separately balloted document. The existing project would be retained as an "umbrella" to handle the coordination issues arising from the split. This is necessary if the parts are to form a coherent whole. New projects would be raised to cover a first set of functional areas. No more than two or three of these functional sub-projects would be active at any time. This would keep the group focussed on a set of limited and achievable goals. New projects would be instantiated as existing ones move into the balloting phase.

One of the benefits of this approach is that each of the new sub-projects must pass the PMC's project approval criteria before it is recommended. The proposal will be properly scrutinized to ensure that the project is likely to succeed within reasonable timeframes. A result of the earlier decision to concentrate on managed objects and management tasks will be to relate the new projects much more closely to existing interfaces, thus removing one of the rods which the group had fashioned for others to beat it with. An obvious source of candidate management tasks can be found in the existing administrative command set on the systems around us, and it would be a perverse decision indeed to introduce gratuitous changes to the style of that interface.

The first set of sub-projects are likely to be Print Management, Software Management, and User Environment Management. These three

represent areas where the work of the group is well advanced and where there is strong commitment of energies.

The Print Management work is based on the MIT Palladium printing system, which has the benefit of being well-aligned with the emerging ISO distributed printing standard, DIS 10164. The Print Management sub-group within POSIX.7 has been working with the Palladium documents for over a year and this work is probably the closest to being complete.

Software Management has enjoyed a resurgence of interest within POSIX.7 over the last 6 months, with source material being drawn from DEC, HP, AT&T, and Siemens-Nixdorf. The small group that has been working in this area has been comparing the various technologies and (not surprisingly?) finding a great deal of commonality between then in terms of their underlying concepts and functionality. The task of identifying a common model and a common set of functions is well advanced and bodes well for the future. (Indeed, the rate of progress is positively alarming!)

The third area, User Environment Management is a logical candidate for inclusion in the initial set of sub-projects. Much of systems management is concerned with the management of users and their interactions with other components of the system. Many management tasks need to be able to refer to users and it seems to be appropriate to tackle this area at an early stage. (For some inexplicable reason, the "add user" operation seems to be the universal example always brought up when talking about some aspect of systems administration - another motivating factor.)

Looking beyond the confines of POSIX.7 into the wider world, the original decision to adopt an object-oriented approach to the problem of systems administration is at last being vindicated. Object-oriented concepts lie at the heart of the OSF Distributed Management Environment request for technology (RFT), the UI Systems Management SIG, and the X/open Systems Management working group. It looks as if history will show POSIX.7's decision to have been a far-sighted move rather than turning up a blind alley.

Report on 1003.9: POSIX Fortran-77 Bindings

E. Loren Buhle, Jr., Ph.D.
<buhle@xrt.upenn.edu> reports on the April 15–19, 1991 meeting in Chicago, IL:

POSIX.9 met to resolve objections and comments raised to the first ballot of the FORTRAN binding to ISO/IEC 9945–1 Standard (also known as POSIX.1). The ballot began in late December 1990 and ended on February 20, 1991. This first proposal did not obtain the necessary 75% acceptance of the balloters. There were 73 people in the total balloting group, of which 56 were eligible to vote on the standard. The others were parties of interest. Of the official balloting group, there were 23 affirmative votes, 15 negative votes, and 8 abstentions. This 82% response was only 60% affirmative. Thus the first ballot failed to make the existing draft a standard.

At the Chicago meeting, objections and comments from all voters (both official and unofficial) were reviewed and acted upon. Many valid points were made by the voters, resulting in changes to the draft. Some revisions included changing the F77 prefixes to PXF (e.g. F77WAIT became PXF-WAIT). Joseph King's request for a "fast exit" was also added.

Fast exit was added back to the draft to gain the _ exit() functionality contained in POSIX.1. It is required to allow proper recovery from failed calls to any of the PXFEXEC() functions within a child process. It seems that recovery means that the child process must be able to exit without flushing buffers. The file buffers of a child process are copies of the parent's. The current draft says that on failure when PXFEXIT(), STOP, and END are executed, the data in the buffers will be written to the file and the child will terminate. So when the parent writes or closes the file, the output buffers will be flushed and data will be duplicated (once from the failed child and once from the parent) in the file.

Most of the objections and comments were resolved in a positive fashion, providing for the possibility of a successful second ballot. With some fast work from the 8 attendees to the POSIX.9 meeting, the revised draft may be recirculated in June for a 30 day period. If all goes

well, the results of the recirculation ballot can be ready for resolution during the July meeting.

The next meeting of the POSIX.9 working group will be July 8–12, 1991 at the Doubletree in Santa Clara, California. The subsequent meeting will be October 21–25, 1991 in Parsippany, NJ.

Report on 1003.12: Protocol Independent Interfaces

Mike Karels <karels@cs.berkeley.edu> reports on the April 15–19, 1991 meeting in Chicago, IL:

Summary

The POSIX.12 (Protocol Independent Interfaces, PII) working group spent the April meeting planning strategy for its new direction and coordinating with other groups. The group will produce a standard encompassing both the BSD sockets and X/open Transport Interface (XTI). Liaison meetings were held with X/open representatives, the Name Space/Directory Services group (POSIX.17) and the Real-Time group (POSIX.4). The group discussed language independent specification issues with Paul Rabin.

Report

POSIX.12 (Protocol Independent Interfaces, PII) spent the April meeting adjusting to its new direction and coordinating with other groups. At the last meeting, the group decided to abandon its previous strategy of producing a new Detailed Network Interface (DNI) with the best features of both the socket and X/open Transport interfaces (XTI). XTI is derived from AT&T's Transport Level Interface (TLI). After reviewing input from users and vendors, the group decided instead to produce a standard including both existing interfaces. In addition, the standard will include the Simple Network Interface (SNI), which would insulate the programmer from lower-level details.

The April meeting included discussions of the changes or additions that were needed for the existing interfaces to become standards. A poll had been sent to several mailing lists and news groups, but few concrete suggestions were received. Most of the suggestions for extensions have come from inside the working group. Suggestions for changes in sockets have come mostly

from the Berkeley representatives, and suggestions for XTI have come mainly from people active in the X/open technical community.

A fair amount of time was devoted to the proposal for extending XTI option management by Gerhard Kieselmann. The proposal allows much more flexible option management by encoding option values with types and lengths. The encoding is similar to the encoding of ancillary data in the 4.3-Reno send and receive calls. The main point of contention was whether the transport provider should maintain both current settings and default settings to be used for any future connections.

The discussions of extensions to the socket interface was confined to a description of the recent Berkeley changes (4.3-Reno) to the socket interface.

The meeting schedule was nearly filled with coordination meetings with other groups. Petr Janacek of X/open reported on the status of future XTI specifications. Other than the option management proposal mentioned above, the XPG4 version of XTI has been finalized. It is hoped that the XPG5 version of XTI will be aligned with the POSIX version. At the last meeting, POSIX.12 asked X/open for editorial assistance in producing a POSIX version of XTI. Petr replied that the budget did not allow for assistance at this time, but that an on-line version of XTI would be made available.

Paul Rabin met with the PII group to discuss issues surrounding POSIX language independent specifications. The working group currently hopes to produce a single language independent specification for DNI; there would be two C language bindings, namely sockets and XTI. This should prevent the necessity of providing two interfaces for languages other than C, but makes the language independent specification more difficult to produce.

The POSIX.12 group also met with members of the Name Space/Directory Services group (POSIX.17) to discuss the DNI dependency on the Directory Services interfaces. There are some problems in this area. The NS/DS group currently intends to provide an interface only to the X.500 directory service, while the PII group assumes an interface that could include other services such as

the Internet Domain Name System. The NS/DS group intends to provide a full-featured low-level interface to the directory service based on the X/open X.500 API. However, they also plan to include simplified higher-level interfaces to answer needs such as this one.

The final coordination meetings were with members of the Real-Time group. The current Real-Time draft includes an interprocess communication (IPC) facility that many believe is too complex and does not extend gracefully to handle networked systems. Many hoped that the IPC interface could be replaced by the 1003.12 interface, with real-time extensions as necessary. A group is working on a straw-man proposal in time for the July meeting.

Report on POSIX.17 - Name Space/ Directory Services

Mark Hazzard <markh@rsvl.unisys.com> reports on the meeting in Chicago, IL:

Summary

The POSIX.17 group is generating a user to directory services API, for example an API to an X.500 Directory User Agent (DUA). We are referring to a network idea of a directory, not the "file which contains file entries" defined in POSIX.1. It is not limited to just the X.500 functionality. We are using XAPIA — X/open's Directory Services specification (XDS) — as a basis for work. XDS is an object-oriented interface and requires a companion specification for object management (XOM).

XOM is a stand alone specification with general applicability beyond the API to directory services. It will be used by IEEE 1224.1 (X.400 API) and possibly other POSIX groups, and is being standardized by IEEE 1224.

We made significant progress on a third draft of the document in Chicago, with the language independent specification work still to be done. We hope to mock ballot the document sometime after the July working group meeting. POSIX.12 (Protocol Independent Interfaces) and POSIX.17 worked together this week and arrived at a number of scenarios for coordinating the work. POSIX.17 is taking steps to determine if their work

overlaps with the proposed work of certain ISO/ SC21 (OSI) working groups.

Status

Commitment within the group remains adequate, but there's more than enough work to go around.

Chris Harding, (from X/open) our Technical Editor, brought a second draft of the specification to the meeting. We made significant progress towards producing a third draft with emphasis on format cleanup, model, overview sections and test assertions.

The "homework" assignments on Language Independent Specification (LIS) weren't completed and additional work on LIS was put on hold until the outcome of the SEC meeting. There seemed to be some confusion as to the applicability of the LIS requirement for POSIX.17 and other Distributed Services APIs. The SEC reaffirmed the LIS requirement. The LIS work was reassigned to the Technical Editor.

The big debate on generalizing the Object Management API never materialized. (Refer to the three snitch reports on the New Orleans 1991 meeting.) I strongly suspect this was largely due to the absence of Scott "Owls in the bushes" Guthrey at the Chicago meeting.

Requirements from POSIX.12

The group met with POSIX.12 (Protocol Independent Interfaces) to get their requirements for the POSIX.17 API. They expressed the desire (necessity?) to:

—access existing directory services (e.g. DNS) via the POSIX.17 API

— map the existing BSD API (e.g. `gethostbyname`, `getservbyname`, etc.) onto the POSIX.17 API.

We discussed at length how these and other requirements should best be met, and produced three different scenarios describing relationships between the user application, the directory API(s), the directory service(s), and the transport service (accessed via POSIX.12's Simplified Network Interface).

In the first scenario, the transport provider (SNI) would talk directly to all directory services

e.g. DNS, X.500, etc. Each directory service resolver would be accessed through its native interface, of which POSIX.17 would be just another API.

In scenario two, POSIX.17 would be the only API and would be used to access all directory services. To access a non-X.500 DUA, the underlying implementation might have to translate POSIX.17 calls into the appropriate format and invoke the corresponding resolver.

In the final scenario, POSIX.17 would again be the only API, but only one resolver (X.500 DUA) would be used to query a single composite information base (DIB) containing information on all objects (e.g. DNS Resource Records and X.500 Distinguished Names).

In each of the scenarios, impact to the POSIX.17 API will be minimal. However, significant impact is anticipated for the underlying implementation and directory information base.

We discussed the relative merits of each and decided that at some future time a single API, resolver (agent), directory service, and information base just might be the best for POSIX systems. We also recognized that POSIX systems will need to interoperate with non-POSIX systems for the foreseeable future, and that fact won't be lost on implementors.

Live long and prosper! or Extending the life of our standard

The base document defines both the API and the collection of objects managed through the API, called a "package." We believe that packages will be much more dynamic than the API itself, and could be unbundled from the API to give the API greater stability. We asked the Distributed Services Steering Committee (DSSC) to recommend a common solution, as this problem is shared by other networking groups. We expect the DSSC to take this issue up in Santa Clara.

Mock Ballot

We decided to try to mock ballot our document sometime after the July meeting. After reaching agreement on the minimum document content for mock ballot, we assigned actions to

get this work done. We wish to solicit input on requirements and feedback on our LIS and Test Assertion work.

Is SC21 doing APIs too?

With the granting of any IEEE project request (PAR) comes a responsibility to coordinate with other de jure standards bodies, the list of which is included on the PAR itself. In fulfilling this obligation, the group has learned (and dutifully reported to the SEC) that ISO SC21 is considering working on APIs to OSI application level services. This work has a potential to overlap the SC22 supported work being done by IEEE TCOS/POSIX (e.g. POSIX.17, P1224, P1238).

In Closing

The group made good solid progress in Chicago, and our document is beginning to flesh out. We think we understand what's required for test assertions and language independence, and have done several things to make the base document more readable. If we can maintain critical mass within the group, we have a good chance of going to mock ballot yet this year. There's a lot of work to do, so we hope you can make it to Santa Clara in July.

Report on P1224: X.400 API

Steve Trus <trus@osi.ncsl.nist.gov> reports on the April 15–19, 1991 meeting in Chicago, IL:

Introduction

P1224 is the IEEE working group standardizing an application program interface (API) for X.400 and also for a companion, OSI Object Management (OM). The work will result in two documents. Interfaces developed by the X.400 API Association and X/Open have provided the basis for the standards. The X.400 API consists of two parts: an application interface and a gateway interface. Both of these are based on the 1988 CCITT X.400 Series of Recommendations.

The P1224 working group has the following officers:

—Steve Trus, Chairman (NIST)

—Tim Carter, Vice Chairman (IBM)

—Iain Devine, Technical Editor, Secretary (X/Open)

The Chicago meeting was very productive for the P1224 working group. We have been gaining momentum over the past three meetings, and are well under way to producing an IEEE standard.

The goal of the group is to have a draft of the X.400 API and the Object Management APIs by the July meeting, and to ballot the documents after the October meeting.

Report

At the Chicago meeting the group continued modifying the base documents to produce the draft API documents for ballot. This work includes:

1. editing the documents to meet the style and format requirements of the IEEE,

2. adding a language independent specification of the interfaces to the documents, and

3. developing the required conformance test assertions.

The language independent specification of the Object Management API is complete, and the technical editor has made most of the required style changes. These changes will be complete and the language independent specification will be incorporated into the document by the July meeting. Work on the style modifications to the X.400 document will also be complete by the July meeting. The X.400 language independent specification should be complete and incorporated at this time.

The group spent most of the week developing the required test methods for the Object Management Specification. A representative of the Test Methods working group (POSIX.3) assisted us with this development. Members of the group agreed to develop test methods for functions assigned to them by the next meeting. This task will need to be completed before the complete ballot of the document.

Balloting Plans

We discussed balloting plans and we would like to begin balloting the Object Management Specification and the X.400 API in October. These ballots would not include the test methods, and balloting cannot complete without them.

We are developing the list of people who will be invited to ballot these documents, along with the IEEE-formed balloting group. This list will include the X.400 API Association, X/Open Limited, the NIST X.400 Workshop, and the Electronic Mail Association.

PAR Restructuring

The original Project Authorization Request (PAR) for the P1224 group was written when the baseline document contained an X.400 gateway API and the related OSI Object Management specification. Currently, the X.400 API document contains the user agent interfaces and the gateway interfaces. The OSI Object Management specification is contained in a separate document. To accommodate these changes a revised PAR was written at the January meeting for the X.400 API, and a new PAR was written for the OSI Object Management specification. These PARs were approved by the IEEE TCOS SEC at this meeting.

In Closing

P1224 is making good progress. Homework assignments were delegated at the Chicago meeting to be completed by the Santa Clara meeting. The primary focus of the Santa Clara meeting will be to review the Draft X.400 and Object Management APIs, and to continue working on test methods for the interfaces.

Report on X3J16: C++

Mike Vilot <mjv@objects.mv.com> reports on the March 1991 meeting in Nashua, NH:

Current Status

The ANSI X3J16 committee began its second year of technical meetings. As expected, the work grew more detailed, with the Core Language and Environment working groups being the focus of most of X3J16's work.

March meeting

Digital Equipment hosted the Nashua meeting. The week's major activities focused on understanding the myriad details of the proposed clarifications and changes to the current working document.

X3J16's sub-groups focused on the key topics listed in the goals statement developed at the

March, 1990 meeting. They worked by electronic mail between meetings, and reported their progress.

International Concerns

Steve Carter, of Bellcore, presented the major international concerns.

Due to the concerns expressed at the November meeting about conversion to a Type I (international) X3 process, Steve came prepared with material explaining the implications of the change. To all appearances, the change seems benign to the technical work of the committee. The change would have the positive effect of getting international involvement. It has the potential to delay the development of the standard, due to the need to synchronize U.S. and ISO balloting.

The full X3J16 committee almost decided to vote to adopt the change, but ran out of the quorum necessary to pass the motion on Friday morning.

Editorial

Jonathan Shopiro, of AT&T, presented the Editorial group's work.

The most significant change from the November version was the incorporation of the exception handling proposal. Jonathan also described an editorial change that simplified the treatment of names and name lookup, merging the concepts that had previously been treated under the topics of dominance and name hiding. Martin O'Riordan, of Microsoft, questioned whether this was a purely editorial change, or a change to the language semantics. Martin and others requested time to look over the change before agreeing to it.

As I mentioned last time, the person who volunteered to edit the Rationale document has not been heard from since last summer. Susan Waggoner, of USWest, has taken on that responsibility.

Formal Syntax

James Roskind, an independent consultant, presented the work of the Formal Syntax group.

The bulk of the discussion concerned a proposal by Reg Charney of Program Conversions,

Inc. to rename the non-terminals in the grammar. Although there was much discussion about the virtues of regularizing the naming versus the evils of gratuitous changes, the committee decided, in the end, to adopt the proposal.

Eric Krohn, of Bellcore, presented the syntactic ambiguities involving the newly-adopted throw-expression syntax for exceptions. The discussion clarified the issues, and a final resolution is likely next meeting.

Tom Penello, of Metaware, gave an interesting presentation on the inherent problems with ambiguous grammars. He established the fact that an ambigous grammar makes the question of a conforming implementation undecidable. He also illustrated that arbitrary rules to resolve grammatical ambiguities has the side-effect of rejecting valid programs.

He then went on to explain the syntactic ambiguities of the template syntax, arising from the conflict over using the ">" symbol as both a relational operator and a template argument list delimiter. Although he proposed a grammar rewrite that solved the problem, he decided not to recommend it on aesthetic grounds.

There seems to be an appreciation within X3J16 as a whole for the technical issues involved in making the grammar correct. There also seems to be a sentiment in favor of letting the semantic rules settle most of the complex issues.

Core Language

Andy Koenig, of AT&T, presented the Core Language group's work.

Document X3J16/91–0005 describes the group's discussion about the linkage of typedef names and anonymous classes. The group decided it was an Environmental issue, and handed it off to the Environment group.

The group discussed objects created under a condition, and resolved to consider those objects governed by an implicit block scope, as if the programmer had explicitly supplied a compound statement. Discussion is summarized in X3J16/91–0021.

Document 91–0019 covers the discussion of lifetimes for temporary objects created by the

compiler. This issue has not reached closure, although the issues were clarified.

Environment

Peter Chapin, of Vermont Technical College, presented the work of the Environment group.

Document X3J16/91–0011 describes the group's discussion about c/c++ compatibility issues. This discussion is continuing.

The group discussed at length the one definition rule — enforcing the rule that a program must have exactly one definition for a given function, even in the presence of multiple inclusions of inline functions and the potential need for the compiler to generate such functions out of line. Document X3J16/91–0024 summarizes the discussion.

There is a proposal to include a section in the standard on required warnings. Laura Yaker, now at Mentor Graphics, presented some ideas of the sorts of things that might be considered as required warnings. The discussion indicated that this is a difficult issue to standardize, since there is so much variation in environments and implementations. This ongoing discussion is summarized in X3J16/91–0014.

Another ongoing discussion concerns static initialization order for objects in different translation units. Document X3J16/91–0012 summarizes this discussion.

There was some discussion on specifying translation limits in the standard. The discussion seemed to generate more heat than light, and nothing was decided.

Lastly, the linkage of types discussion continues, and is summarized in X3J16/91–0023. Peter described several alternate rules to ensure type-safe linkage of types. A central issue is whether the linkage specification is part of the type. There are interesting arguments for and against this.

Libraries

I presented the Library group's work.

There has been some progress on formulating proposals for submission to X3J16. Aron Insinga of DEC presented his proposal to apply templates

to the definition of the standard string class. His progress has been slowed by the lack of an available implementation supporting templates.

Steve Clamage of TauMetric presented proposed resolutions for almost all of the compatibility issues regarding the C library. Most of the small type insecurities can be handled in a reasonably straightforward manner. There are more substantial issues regarding signals, exceptions and the facilities provided by longjmp().

The iostreams proposal continues to receive comment. Many of the UNIX-specific issues have been removed. Addressing these concerns raised an interesting point — should the C++ standard adopt the practice of the C standard, in describing only that certain types exist, or should it describe them as classes and specify their required operations? There was some concern that describing classes would be inefficient, but other concerns that the vague wording without a class description would introduce too much variability among implementations.

Language Extensions

Bjarne Stroustrup, of AT&T, presented the work of the Extensions group.

The group is working through a long list of proposals for changes to the language. A significant number of them came from the Core language group, due to an evaluation of what Andy Koenig calls "language extension by technicality" — where suggestions for changing the wording of the standard would have the effect of changing the meaning of the language.

The current list of language extension proposals includes overloading of the "." operator, a proposal for handling national character set issues with digraphs and new keywords, and the adoption of the "inherited" keyword (as in Apple's implementation).

The largest issue lurking in the Extensions category is the addition of support for run-time type information. There will be much discussion on this topic over the next months.

C Compatibility

Tom Plum, of Plum-Hall, presented the work of the C Compatibility group.

The group continued its investigation of the vocabulary differences between C and C++. They decided to categorize their efforts into groups, covering the language, environment, and library. One likely outcome of their work will be a proposal to adopt the same model of sequence points used by X3J11.

Next events

The next three X3J16 meetings (and their hosts) will be:

- June 17–21, Lund, Sweden (Lund Institute of Technology)

- November 11–15, Toronto, Canada (IBM)

- March 1992, Austin, TX (TI)

Zortech announced plans to host one of the other two 1992 meetings in London.

Membership on an X3 committee is open to any individual or organization with expertise and material interest in the topic addressed by the committee. The cost for membership is $250. Contact the chair or vice chair for details.

Chair:
Dmitry Lenkov
HP California Language Lab
19447 Pruneridge Avenue MS 47 LE
Cupertino, CA 95014
(408)447–5279
FAX
(408)447–4924
email *dmitry%hpda@hplabs.hp.com*

Vice Chair:
William M. Miller
Glockenspiel, Ltd
P.O. Box 366
Sudbury, MA 01776–0003
(508)443–5779
email *wmmiller@cup.portal.com*

If you are a member of DECUS, the following information is of interest to you.

DECUS is taking an increasingly active role in POSIX, and received institutional representative status to the TCOS-SS SEC at this meeting. If you wish to express an opinion or ask a question regarding specific POSIX subgroups, they should be directed towards your DECUS representatives:

DECUS	
REPRESENTATIVE	ALTERNATE
Joseph J. King, Ph.D. Genetics Computer Group 575 Science Drive, Suite B Madison, Wisconsin, 52711 Jking@gcg.com (608) 231-5200 DCS: KING	E. Loren Buhle, Jr. Ph.D. University of Pennsylvania Rm 440A, 3401 Walnut St. Philadelphia, PA, 19104 buhle@xrt.upenn.edu (215) 662-3084 DCS: BUHLE

Book Reviews

POSIX Programmer's Guide
by Donald Lewine

(O'Reilly & Associates, 1991, ISBN 0–937175–73–0, $34.95 597 pages)

The POSIX.1 Standard, A Programmer's Guide
by Fred Zlotnick

(Benjamin/Cummings, 1991, ISBN 0–8053–9605–5, $23.96, 379 pages)

Reviewed by Peter Collinson
Hillside Systems

Standards are not intended to be *readable* documents; ease of reading is sacrificed in favour of other attributes. A standard should be as concise, unambiguous, and accurate as possible to express how something should happen. It's good to see a couple of books that take the POSIX.1 standard and set out to explain how it is intended to be used. The standard defines the system interface for programs running on a POSIX conforming system, the full title of which is ISO/IEC 9945–1, Information Technology — Portable Operating System Interface (POSIX) — Part 1: System Application Program Interface (API)[C Language]. The "C language" in brackets shows that the standard is written in terms of C and is currently inextricably tied to the ANSI C standard. As a result both books also cover large parts of the C standard.

As a programmer what am I looking for in a POSIX book? First, I would like some background information to the standard. Then I need a description of how the standard affects the environment in which my programs will run. This is more than just the environment strings. It's also the notions of header files, how my program is able to get information about the system, and whether the system supports the 4.3BSD notion of *chown* or whether it treats ownership like System V.3 does.

I then need to know how files and directories work, how access permissions are defined, the basic set of system calls to deal with files, how devices are connected into the file system, and how pipes and fifos work. Since POSIX defines a set of function calls not a million miles away from the Portable C library, I would expect to see some discussion of that. Since programs will run in processes, I expect to find some discussion on process creation and synchronisation. Since the POSIX standard is dependent on the ANSI-C standard, I would also want to see some discussion of the routines that the C standard has placed into the world.

There are then some big areas that I would wish to read. Signals is an area where the worlds of BSD and System V have diverged. POSIX attempts to reconcile these differences. We all need to know how signals have been redefined. If you come from the BSD world, then the way that terminals are controlled has completely altered. Many familiar concepts dating back to the earliest UNIX systems have disappeared to be replaced by others. The areas of job control and sessions should also be covered.

Finally, it would also be good to see a comprehensive reference section so that the book would continue to be useful when programming.

Donald Lewine's book certainly covers all this material. The book is split into two sections, the first third containing ten chapters of discussion. This is followed by 300 pages containing the descriptions of the library functions that a programmer will use. The second part of the book is done well, and will be very useful as a reference guide. I especially liked the addition of references to the standards on each "manual" page. There are several appendices and a comprehensive index.

I think that the first section of the book is adequate but very thin in many places. For example, the chapter on processes has a lot of discussion on the system calls that exist and only a small amount giving the why and the how. The book avoids much mention of sessions and job control. This is relegated to a couple of pages at the end of the chapter on terminals. The book really fails when things get more technical.

However, it is well written and is easy to read. (I am saying this even though the first few chapters are plastered with one of my pet hates — footnotes! Please can people stop writing with footnotes? I always read them in-line so why not put the text in-line?)

Fred Zlotnick's book, *The POSIX.1 Standard* is much more complete from a technical point of view. The book is much denser and possibly harder to read. It gives a lot more background and explanation of why decisions were taken, and does more comparison between existing reference systems System V.3 and 4.3BSD. I like the way that every routine specification is shown with any associated header files.

It covers all the material that I mentioned above and also contains sections on Data Interchange formats, that's *cpio*, *tar*, and *pax*. There is a chapter on the revisions to the POSIX.1 standard and a chapter on related standards. The book ends with appendices containing POSIX functions, ANSI C functions, error numbers, and headers.

I much prefer Zlotnick's book. It seems more complete and contains more information. However, the reference material is not as good as Lewine's book and I would guess that if I started programming to the standard then Lewine's book would lie about the desk being used as reference material.

Request for Proposals to Chair the Large Installation Systems Administration (LISA) Conference

The USENIX Association is once again seeking proposals from people interested in chairing its sixth LISA conference, to be held sometime in the Fall of 1992.

We are seeking an energetic person with the following qualifications:

- Good administrative skills
- A lot of experience in the administration of large installations
- Good public speaking skills
- Knowledge of what are the timely/ appropriate topics in the field
- Ability to solicit good panel members/ appropriate speakers
- Attendance at previous LISA workshops/ conferences

Proposals should be brief (1 page) and might include the following:

- Statement of Purpose (e.g., why should we have another one?)

- Form of submissions (e.g., abstracts, extended abstracts or full papers?)
- Format (e.g., 2 or 3 days of technical sessions, panel sessions, etc.)
- List of topics to be addressed, as in the call for papers
- Special features, such as having tutorials, BOFs, vendor demos
- List of potential program committee members and/or a co-chair*
- Biography and references

*While most USENIX conferences have had an individual chair, proposals requesting a co-chair and/or small program committee are welcome.

Proposal due date: **October 9, 1991.**

Please address all inquiries and proposals to the Association's exectuve director, Ellie Young (*ellie@usenix.org*).

ORDER FORM

CONFERENCE & WORKSHOP PROCEEDINGS

Qty	Proceedings		Member Price	Non-Member Price	Total	Foreign Postage	Total
__	Nashville Conference	June '91	$32	$38	$_____	$22	$_____
__	Dallas Conference	Jan. '91	28	34	$_____	18	$_____
__	Anaheim Conference	June '90	22	22	$_____	15	$_____
__	Washington DC Conference	Jan. '90	25	25	$_____	15	$_____
__	Baltimore Conference	June '89	20	20	$_____	15	$_____
__	San Diego Conference	Feb. '89	30	30	$_____	20	$_____
__	Distributed & Multiprocessor Sys. (SEDMS) II	Mar '91	30	36	$_____	20	$_____
__	Distributed & Multiprocessor Sys. Workshop	Oct. '89	30	30	$_____	20	$_____
__	Mach Workshop	Oct. '90	17	20	$_____	9	$_____
__	Large Installation Sys. Admin. IV Conference	Oct. '90	15	18	$_____	8	$_____
__	UNIX Security II Workshop	Aug.'90	13	16	$_____	8	$_____
__	C++ Conference	Apr. '91	22	26	$_____	11	$_____
__	C++ Conference	Apr. '90	28	28	$_____	18	$_____
__	C++ Conference	Oct.'88	30	30	$_____	20	$_____
__	C++ Workshop	Nov.'87	30	30	$_____	20	$_____
__	UNIX Transaction Processing Workshop	May '89	12	12	$_____	8	$_____
__	Software Management Workshop	Apr. '89	20	20	$_____	15	$_____
__	UNIX and Supercomputers Workshop	Sept. '88	20	20	$_____	10	$_____
__	Graphics Workshop V	Nov. '89	18	18	$_____	10	$_____
__	Graphics Workshop IV	Oct. '87	10	10	$_____	15	$_____
__	Washington DC Conference	Jan. '87	10	10	$_____	20	$_____

Total price of Proceedings _____
Calif. residents add 8.25% sales tax _____
Total foreign postage _____
Total enclosed $_____

Discounts are available for bulk orders. Please inquire.

PAYMENT OPTIONS

☐ Check enclosed payable to USENIX Association.   ☐ Purchase order enclosed.*

☐ Please charge my: ☐ Visa ☐ MasterCard [VISA] [●●]

Account # _____ Exp.Date _____

Signature _____

* Outside the USA? Please make your payment in U.S. currency by one of the following:
   * Charge (Visa, MasterCard, or foreign equivalent)
   * International postal money order
   * Check – issued by a local branch of a U.S. Bank

7/91

Shipping Information

Orders to U.S. and Canada are shipped via printed matter. Please allow 2 – 3 weeks for delivery. Overseas orders are shipped via air printed matter.

Ship to: _____   Please mail this order form to: USENIX Association
                                                                          2560 Ninth Street
          _____   Suite 215
                                                                          Berkeley, CA 94710
          _____

          _____

Local User Groups

The Association will support local user groups by doing a mailing to assist the formation of a new group and publishing information on local groups in *;login:*. At least one member of the group must be a current member of the Association. Send additions and corrections to *login@usenix.org*.

CA – Fresno: the Central California UNIX Users Group consists of a *uucp*-based electronic mailing list to which members may post questions or information. For connection information:

Educational and governmental institutions:
Brent Auernheimer   (209) 278-4636
brent@CSUFresno.edu or csufres!brent

Commercial institutions or individuals:
Gordon Crumal   (209) 251-2648
csufres!gordon

CA – Irvine: the UNIX Users Association of Southern California meets the 2nd Monday of each month.

Rich Bergstedt   (714) 582-0768
26755 Dulcinea
Mission Viejo, CA 92691
attmail.com!bergstedt

CO – Boulder: the Front Range UNIX Users Group meets monthly at different sites.

Steve Gaede   gaede@sda.com
Software Design   (303) 444-9100
   & Analysis, Inc.
1113 Spruce St., Ste. 500
Boulder, CO 80302

FL – Coral Springs:

S. Shaw McQuinn   (305) 344-8686
8557 W. Sample Road
Coral Springs, FL 33065

FL – Fort Lauderdale/Miami: The South Florida UNIX Users Group meets the 2nd Tuesday of each month.

Tony Vincent, John McLaughlin   (305) 776-7770
{sun,novavax,gould}!sunvice!tony
jmclaughlin@sun.com

John O'Brien   (305) 475-7633
gatech!uflorida!novavax!john

FL — Western: The Florida West Coast UNIX Users Groups meeting the first Thursday of every month

Richard Martino   (813) 536-1776

Tony Becker   (813) 799-1836
mcrsys!tony

Ed Gallizzi, Ph.D.   (813) 864-8272
e.galizzi@comtmail.com

Jay Ts   (813) 979-9169
uunet!pdn!tscs!metran!jay

Bill Davis   (407) 242-4449
bill@ccd.harris.com

FL – Orlando: the Central Florida UNIX Users Group meets the 3rd Thursday of each month.

Mike Geldner   (407) 862-0949
codas!sunfla!mike

Ben Goldfarb   (407) 275-2790
goldfarb@hcx9.ucf.edu

Mikel Manitius   (407) 869-2462
(codas,attmail)!mikel

FL – Tampa Bay: the Tampa UNIX Users Group meets the 1st Thursday of each month in Largo.

Bill Hargen   (813) 530-8655
uunet!pdn!hargen

George W. Leach   (813) 530-2376
uunet!pdn!reggie

GA – Atlanta: meets on the 1st Monday of each month in White Hall, Emory University.

Atlanta UNIX Users Group
P.O. Box 12241
Atlanta, GA 30355-2241

Mark Landry   (404) 365-8108

MI – Detroit/Ann Arbor: The SouthEastern Michigan Sun Local Users Group meets jointly with the Nameless UNIX Group on the 2nd Thursday of each month in Ann Arbor.

Steve Simmons   home: (313) 426-8981
scs@lokkur.dexter.mi.us   office: (313) 769-4086

K. Richard McGill   Bill Bulley
rich@sendai.ann-arbor.mi.us   web@applga.uucp

MI – Detroit/Ann Arbor: dinner meetings the 1st Wednesday of each month.

Linda Mason   (313) 855-4220
michigan!/usr/group
P.O. Box 189602
Farmington Hills, MI 48018-9602

MN – Minneapolis/St. Paul: meets the 1st Wednesday of each month.

UNIX Users of Minnesota   Robert A. Monio
17130 Jordan Court   pnessutt@dmshq.mn.org
Lakeville, MN 55044   (612) 220-2427

MO – St. Louis:

St. Louis UNIX Users Group	Eric Kiebler
Plus Five Computer Services	plus5!sluug
765 Westwood, 10A	(314) 725-9492
Clayton, MO 63105	

NE – Omaha: meets monthly.

/usr/group/nebraska	
P.O. Box 31012	Philip Allendorfer
Omaha, NE 68132	(402) 423-1400

New England – Northern: meets monthly at different sites.

Peter Schmitt	
Peter.Schmitt@dartvax!dartmouth.edu	
Kiewit Computation Center	(603) 646-2085
Dartmouth College	
Hanover, NH 03755	

NJ – Princeton: the Princeton UNIX Users Group meets monthly.

Peter J. Holsberg	mccc!pjh
Mercer County	(609) 586-4800
Community College	
1200 Old Trenton Road	
Trenton, NJ 08690	

NY – New York City: Unigroup of New York City meets every other month in Manhattan.

Unigroup of New York City	
G.P.O. Box 1931	
New York, NY 10116	
Peter Gutmann	(212) 618-0973
peterg@murphy.com	

OH – Columbus: The Columbus Local UNIX Group meets the 1st Monday of each month.

Mark Verber	verber@mps.ohio-state.edu
Physics Department	(614) 292-8002
Ohio State University	
Columbus, OH 43210	

OK – Tulsa: the Tulsa UNIX Users Group, $USR, meets the 2nd Wednesday of each month.

Stan Mason	(918) 560-5329
tulsix!smason@drd.com	
Mark Lawrence	(918) 743-3013
mark@drd.com	

TX – Austin: CACTUS meets the 3rd Thursday of each month.

Capital Area Central Texas UNIX Society	
P.O. Box 9786	
Austin, TX 78766-9786	
officers@cactus.org	
James Johnson	(512) 331-3781
president@cactus.org	

TX – Dallas/Fort Worth:

Dallas/Fort Worth UNIX Users Group	
660 Preston Forest, Suite 177	
Dallas, TX 75230	
Kevin Coyle	(214) 991-5512
kevinc@shared.com	

TX – Houston: the Houston UNIX Users Group (Hounix) meets the 3rd Tuesday of each month.

Hounix answering machine	(713) 684-6590
Bob Marcum, president	(713) 270-8124
Chuck Bentley, vice-president	(713) 789-8928
chuckb@hounix.uucp	

WA – Seattle: meets monthly.

Bill Campbell	(206) 232-4164
Seattle UNIX Group Membership Information	
P.O. Box 820	
Mercer Island, WA 98040-0820	
uw-beaver!tikal!camco!bill	

Washington, D.C.: meets the 1st Tuesday of each month.

Washington Area UNIX Users Group	
9811 Mallard Drive	
Laurel, MD 20708	
Alan Fedder	(301) 953-3626

CANADA – Toronto:

Evan Leibovitch	(416) 452-0504
143 Baronwood Court	evan@telly.on.ca
Brampton, Ont. Canada L6V 3H8	

SEDMS III

Report from Nashville

Book Reviews